



Universitat Politècnica de Catalunya

ESCUELA SUPERIOR DE INGENIERÍAS INDUSTRIAL,
AEROSPAcial Y AUDIOVISUAL DE TERRASA

Grado en Ingeniería en Vehículos Aeroespaciales

ESTUDIO Y DISEÑO DEL CONTROL REMOTO DEL ROBOT EXPLORADOR: ALPHABOT.

—Memoria—

Trabajo de Fin de Grado

Autor: Luna Ariadna Farrero Benítez

Tutor: Jose Antonio Soria Pérez

30 de junio de 2020

Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor, Jose Antonio Soria, por haber tutorizado el proyecto como es debido. Me ha sabido guiar y ayudar en todo momento, enseñándome a aprender de los errores cometidos y arreglándolos. Gracias al grupo de investigación DISEN (*Distributed Sensor Networks*) de la Universitat Politècnica de Catalunya, por haberme facilitado los medios necesarios, a través de mi tutor, para la realización de este Trabajo de Final de Grado, ya que de no ser así, no me habría sido posible disponer de los medios e instalaciones que me han ayudado a desarrollar el proyecto.

También me gustaría agradecer a todos aquellos investigadores que han hecho posible, de manera indirecta, la investigación sobre la temática tratada en mi proyecto. Sin ellos y sus hallazgos no hubiese sido posible el tema elegido.

Por último, pero no menos importante, gracias a mi familia y amigos que me han apoyado en todo momento, ayudándome a superar los momentos difíciles y estresantes a nivel psicológico que un proyecto lleva consigo.

Resumen

Los satélites espaciales han revolucionado nuestro mundo completamente. Explorar los otros planetas del Sistema Solar nos ayuda a comprender el nuestro y conocer las repercusiones que puede tener actuar en él. Un día, la propia existencia de la Tierra y todos los que vivimos en ella, puede depender de saber detectar y desviar un asteroide a una órbita que no suponga un peligro de impacto en la superficie terrestre. Desde el punto de vista de la tecnología, los ordenadores, las comunicaciones, entre otros, se han beneficiado de los desarrollos creados para la investigación espacial, ya que los satélites de comunicaciones son los que ofrecen acceso instantáneo a informaciones de todo el mundo vía radio, televisión y móviles. Por todo ello, la investigación espacial es muy importante y necesaria en la vida humana.

Para llevar a cabo misiones a otros planetas, se desarrollan estructuras móviles robotizadas a las que se le incorporan todas las herramientas e instrumentos necesarios para realizar los estudios requeridos, al mismo tiempo que se les equipa con sistemas de protección para trabajar en condiciones hostiles donde los humanos no pueden llegar.

Con el objetivo de desarrollar una aplicación ambientada en el campo de los vehículos aeroespaciales, este proyecto trata de incentivar el interés por la exploración espacial y la robótica móvil. Por ello, y a modo de ejemplo, se ha desarrollado una interfície web para gestionar el robot móvil Alphabot, de Waveshare, controlando movimientos remotamente, realizando instantáneas y grabaciones de vídeo, así como medir datos de temperatura y humedad.

Palabras clave: Alphabot, robótica móvil, desarrollo web, exploración espacial, control remoto, unidad de control y proceso.

Abstract

Space satellites have completely transformed our world. Exploring the other planets of the Solar System helps us to understand our own and to know the repercussions that acting on it can have. One day, the whole existence of the Earth and all of us who live on it could depend on knowing how to detect and divert an asteroid into an orbit that does not pose an impact risk on the Earth's surface. From the point of view of technology, computers, communications, among others, have benefited from the developments created for space research, since communication satellites are the ones that offer instant access to information from all over the world via radio, television and mobile phones. For all these reasons, space research is very important and necessary in human life.

To be able to carry out missions to other planets, robotic mobile structures are developed to which all the tools and instruments necessary to perform the required studies are incorporated, while at the same time they are equipped with protection systems to work in hostile conditions where humans cannot reach.

With the aim of developing an application set in the field of aerospace vehicles, this project seeks to encourage interest in space exploration and mobile robotics. Therefore, and as an example, a web interface has been developed to manage the Alfabot mobile robot, from Waveshare, controlling movements remotely, taking snapshots and video recordings, as well as measuring temperature and humidity data.

Keywords: Alfabot, mobile robotics, web development, space exploration, remote control, control and process unit

Declaración de honor

Declaro que,

el trabajo en este Trabajo de Fin de Grado es completamente mi trabajo,

ninguna parte de esta tesis de grado se toma del trabajo de otras personas sin darles crédito,

todas las referencias se han citado claramente.

Entiendo que una infracción a esta declaración me deja sujeto a las medidas disciplinarias previstas por la Universitat Politècnica de Catalunya - BarcelonaTECH.

Luna Ariadna Farrero Benítez

Nombre del estudiante

30/06/2020

Firma

Fecha

Título del Trabajo: *Estudio y diseño del control remoto del robot explorador: Alphabot.*

Índice general

| | |
|--------------------------------------------------|------------|
| Agradecimientos | I |
| Resumen | II |
| Abstract | III |
| Declaración de honor | IV |
| Índice de figuras | XI |
| Índice de tablas | XII |
| | |
| 1. Introducción | 1 |
| 1.1. Antecedentes | 1 |
| 1.2. Objetivo del proyecto | 2 |
| 1.3. Alcance del proyecto | 2 |
| 1.4. Especificaciones y Requerimientos | 3 |
| 1.5. Proyectos entregables | 4 |
| 1.6. Motivación de este trabajo | 4 |
| 1.7. Resumen del contenido | 5 |
| | |
| 2. La robótica y el espacio | 6 |
| 2.1. Robótica | 6 |

| | |
|-------------------------------------------------------------|-----------|
| 2.2. Robots móviles | 8 |
| 2.2.1. Mercado de robots móviles | 8 |
| 2.2.2. Aplicaciones de robots móviles | 9 |
| 2.3. Robótica en el ámbito aeroespacial | 12 |
| 2.3.1. Introducción | 12 |
| 2.3.2. Rovers | 14 |
| 2.3.3. Otros robots de apoyo en el espacio | 21 |
| 2.4. Componentes del Alphabot | 22 |
| 2.4.1. Placa controladora | 24 |
| 2.4.2. Sensores | 28 |
| 2.4.3. Actuadores eléctricos | 35 |
| 2.4.4. Otros componentes del AlphaBot | 38 |
| 2.4.5. IoT y desarrollo de aplicaciones remotas | 38 |
| 2.5. Discusión | 42 |
| 2.5.1. Controlador del robot | 43 |
| 2.5.2. Comunicación con el <i>gateway</i> | 44 |
| 2.5.3. Servidor web | 44 |
| 2.5.4. Conclusión | 45 |
| 3. Diseño del control remoto y desarrollo | 47 |
| 3.1. Introducción | 47 |
| 3.2. Entorno del control remoto del robot | 47 |
| 3.3. Configuración de red | 49 |
| 3.4. Configuración básica de la unidad de proceso | 49 |
| 3.5. Diseño de la interfaz web en localhost | 52 |
| 3.5.1. <i>Front-End</i> | 53 |

| | |
|------------------------------------------------------------------|-----------|
| 3.5.2. <i>Back-End</i> | 55 |
| 3.6. Conclusión | 59 |
| 4. Resultados: pruebas, puesta en marcha y funcionamiento | 60 |
| 4.1. Introducción | 60 |
| 4.2. Puesta en marcha | 60 |
| 4.3. Resultados | 62 |
| 4.4. Conclusión | 63 |
| 5. Conclusiones y Trabajo futuro | 64 |
| 5.1. Conclusiones | 64 |
| 5.2. Trabajo futuro | 65 |
| 5.2.1. Mejoras de la interficie | 65 |
| 5.2.2. Mejoras del software/hardware | 66 |

Índice de figuras

| | |
|-------------------------------------------------------------------------------------------------|----|
| 1.1. Estructura de los componentes para la plataforma del control remoto del Alphanbot. | 3 |
| 2.1. Robot colaborativo de ABB: <i>IRB 14050 Single-Arm YuMi</i> | 7 |
| 2.2. Pluto: <i>Penn Laboratory for Underground Tunnel Operations</i> | 7 |
| 2.3. Algunas posibles trayectorias a seguir por el robot móvil. | 8 |
| 2.4. Spot y Atlas, robots de Boston Dynamics. | 9 |
| 2.5. Aspirador doméstico: Roomba de iRobot. | 10 |
| 2.6. Mapa creado por Roomba. | 10 |
| 2.7. Aplicación de iRobot para conocer la actividad del robot. | 10 |
| 2.8. Robot educacional: RoboMaster S1. | 11 |
| 2.9. Menú principal de la app RoboMaster. | 11 |
| 2.10. Robot militar explorador Fedor. | 12 |
| 2.11. La nave Sputnik 1. | 13 |
| 2.12. La Estación Espacial Internacional, ISS. | 13 |
| 2.13. Rover Sojourner de la NASA. | 14 |
| 2.14. Placa solar del Rover Sojourner. | 15 |
| 2.15. Baterías del Rover Sojourner. | 15 |
| 2.16. Hardware del módem de radio Rover. | 16 |
| 2.17. APXS de Mars Sojourner. | 17 |

| | |
|----------------------------------------------------------------------------------|----|
| 2.18. Partes e instrumentos del Rover Sojourner. | 17 |
| 2.19. Rover Curiosity. | 18 |
| 2.20. MMRTG del Curiosity. | 19 |
| 2.21. Transceptor Electra. | 20 |
| 2.22. Curiosity transmite datos a la Tierra. | 20 |
| 2.23. Cámaras del Curiosity. | 21 |
| 2.24. Canadarm. | 22 |
| 2.25. Elementos de la placa base del Alphabot. | 23 |
| 2.26. Elementos que componen un microcontrolador. | 26 |
| 2.27. Placa Raspberry Pi 4B. | 26 |
| 2.28. GPIO de Raspberry Pi marcados de color naranja. | 27 |
| 2.29. Placa Arduino UNO. | 27 |
| 2.30. Pines y partes del Arduino UNO. | 28 |
| 2.31. Sensor fotoeléctrico tipo barrera. | 29 |
| 2.32. Sensor fotoeléctrico tipo reflectivo. | 29 |
| 2.33. Sensor fotoeléctrico tipo retroreflectivo. | 30 |
| 2.34. Circuito acondicionador de detector óptico. | 30 |
| 2.35. Sensor FC-51. | 30 |
| 2.36. Módulo siguelíneas. | 31 |
| 2.37. Conexión eléctrica de los sensores IR. | 31 |
| 2.38. Imagen que ilustra cómo viajan las ondas de un sensor ultrasónico. | 32 |
| 2.39. Sensor HC-SR04. | 32 |
| 2.40. Pulsos cuadrados transmitidos. | 32 |
| 2.41. Sensor DHT11. | 34 |
| 2.42. Trama de datos de bits del DHT11. | 34 |
| 2.43. Sensor WYC-H206. | 35 |

| | |
|------------------------------------------------------------------------------|----|
| 2.44. Encoder del Alphabot. | 35 |
| 2.45. Motores DC del AlphaBot. | 36 |
| 2.46. Interruptores en puente en H. | 36 |
| 2.47. Servomotor SG90. | 37 |
| 2.48. Pulsos PWM del servomotor. | 37 |
| 2.49. Baterías de Li-Ion 18650. | 38 |
| 2.50. Cámara Raspberry Pi para el Alphabot. | 38 |
| 2.51. Conexión entre componentes LAMP. | 41 |
| 2.52. Funcionamiento de las solicitudes web con Django. | 42 |
| 3.1. Arquitectura de la plataforma desarrollada. | 47 |
| 3.2. Página de inicio. | 48 |
| 3.3. IPs asignadas por el enrutador. | 49 |
| 3.4. Putty. | 50 |
| 3.5. Opciones de interfaces de Raspberry Pi. | 51 |
| 3.6. Configuración para conexión escritorio remoto. | 51 |
| 3.7. Conexión al escritorio Raspbian. | 52 |
| 3.8. Flujo de la información entre módulos de la aplicación web | 53 |
| 3.9. Sección para el control remoto de movimientos. | 54 |
| 3.10. Sección que muestra datos de los sensores. | 54 |
| 3.11. Mapa de los pines del Alphabot. | 55 |
| 3.12. Mapa de los puertos GPIO de la RPi en el Alphabot. | 56 |
| 3.13. Puerto del módulo de la cámara. | 57 |
| 3.14. Conexión del sensor DHT11. | 57 |
| 3.15. Diagrama de flujo simplificado del trabajo de la Raspberry Pi. | 58 |
| 4.1. Leds ACT y PWR de Raspberry Pi. | 61 |

| | |
|----------------------------------------------------------------------------------------------|----|
| 4.2. Esquema del sistema de red. | 61 |
| 4.3. Sección de la cámara. | 62 |
| 4.4. Escritorio de Raspbian con los archivos capturados y guardados automáticamente. | 63 |

Índice de tablas

| | |
|------------------------------------------------------------------------------------------------------------|----|
| 2.1. Control teórico de los motores del AlphaBot. | 37 |
| 2.2. Selección de la unidad de control para el proyecto. | 43 |
| 2.3. Selección del servidor web para el desarrollo de la aplicación. | 45 |
| 2.4. Software y hardware que componen el proyecto. | 46 |
| 3.1. Interfaces del Alphabot y su correspondencia respecto los puertos GPIO de la Raspberry Pi. | 56 |
| 3.2. Control teórico de los motores del AlphaBot. | 56 |

Capítulo 1

Introducción

1.1. Antecedentes

Según William Gerstenmaier [1], responsable de Exploración y Operaciones Tripuladas de la NASA, explorar el universo ayuda a desarrollar tecnologías que hoy en día son útiles para el ser humano, sirve para hacer más fácil la vida en la Tierra, y por ello, genera grandes beneficios para la humanidad. Por ejemplo, para la misión Apollo se descubrió la tecnología que actualmente se usa en los móviles, Internet comenzó gracias a ello.

De acuerdo con [2], en la próxima década (2020-2029), la financiación de la exploración espacial crecerá alrededor del 50 % respecto los últimos 10 años, lo cual refleja el apoyo de los gobiernos a nivel mundial para los planes exploradores previstos. Aun teniendo en cuenta que la crisis económica y de salud actual puede afectar la inversión a corto plazo, están seguros de que no lo hará a largo plazo.

Pero para llevar a cabo exploraciones espaciales, donde los entornos son demasiado hostiles para los humanos, se requieren de robots que sean más robustos y lo suficientemente autónomos de acuerdo con la misión planeada.

A medida que pasan los años, el desarrollo en el mundo de la robótica aumenta y se vuelve cada vez más importante y cotidiano en nuestro día a día. Son muchas las empresas que sustituyen el trabajo humano poniendo en su lugar máquinas diseñadas para hacer unas tareas específicas, en menor cantidad de tiempo y sin margen de error. A parte de los robots espaciales, como los Rovers, creados para explorar Marte, hay muchos más ámbitos que optan por esta innovadora tecnología, como los brazos robóticos utilizados en la construcción de automóviles, robots en el sector médico para ayudar en cirugías complicadas, junto con muchas más.

La previsión de crecimiento de los robots para 2021 ha dado un salto cualitativo mayor a raíz de la pandemia de COVID-19, ya que resuelven el problema de la

distancia física, tan necesaria en estos tiempos, en cuanto a la atención sanitaria con las personas. Así pues, se estima una incorporación de 522 mil unidades de robots autónomos en empresas en este próximo año 2021 [3].

1.2. Objetivo del proyecto

El objetivo de este proyecto es desarrollar el control remoto del robot explorador Alfabot, mediante una aplicación web conectada a sus actuadores, sensores y cámara. Por control remoto se entiende el manejo remoto de los movimientos y componentes del robot.

1.3. Alcance del proyecto

Con el fin de lograr el objetivo de este proyecto, los componentes de la estructura que permite controlar remotamente el robot son la CPU, una puerta de enlace y una aplicación web (Fig. 1.1). El comportamiento de estos elementos será el siguiente:

- La unidad de control es la placa controladora de todo el sistema del robot. En ella estará todo el hardware del robot, los códigos programados para el movimiento de éste, el correcto funcionamiento de la cámara y el control de los sensores incorporados. La placa deberá recibir datos en forma de comandos y variables que se manden desde la web para realizar unos movimientos u otros y accionar los sensores y/o la cámara. También se buscará que contenga el servidor web que sirva para alojar la página donde el usuario interacciona para gestionar el robot.
- La puerta de enlace (o '*gateway*') será el puente entre el robot y la aplicación web, la cual será un *router*, permitiendo el diálogo en ambos sentidos de forma inalámbrica, siempre que los dispositivos estén dentro de la misma red.
- La aplicación web permitirá al usuario interactuar con el robot remotamente. Sus operaciones serán manejar el movimiento del robot, encender/apagar la cámara cuando sea necesario, y experimentar con ella, detectar obstáculos mediante sus sensores de infrarrojos, y por último, conocer la temperatura y humedad.
- Las consideraciones a tener en cuenta para soportar condiciones extremas en las misiones de exploración del robot van más allá de este proyecto y no serán estudiadas.

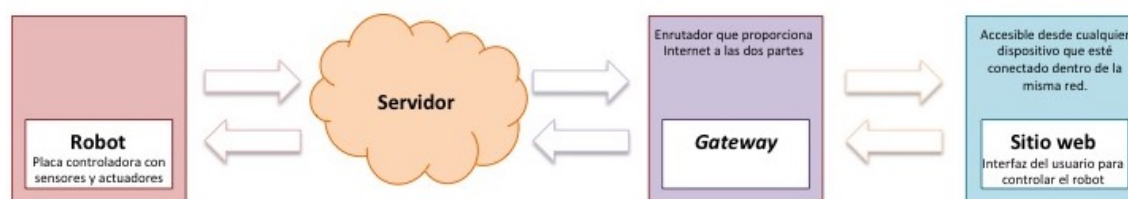


Figura 1.1: Estructura de los componentes para la plataforma del control remoto del Alfabot.

1.4. Especificaciones y Requerimientos

- **Sensores y actuadores:** Son los elementos que constituyen el hardware del TFG. Se ha optado por adquirir la plataforma Alfabot ya que sólo requiere su ensamblaje, más que tener que diseñar las PCBs (*Printed Circuit Board*). Los sensores infrarrojos implementados son capaces de detectar obstáculos que se encuentren en frente del rumbo del robot, permitiendo obtener datos y variables físicas para evaluar la información del entorno que pueda servir para moverse en el mismo. Por otro lado, el sensor de temperatura y humedad lee datos de éstas dos magnitudes físicas en su ambiente.
- **Cámara:** La cámara de 8Mpx controlada directamente por el usuario, permite emitir vídeo en vivo, capturar imágenes y grabar vídeos que se guarden automáticamente en la memoria de la unidad de control.
- **Sistema locomotor:** Incluye todos los elementos encargados de proporcionar el movimiento a las articulaciones del robot que permiten su desplazamiento, así como la plataforma móvil donde se aloja la cámara. Estos elementos son los servomotores instalados en la plataforma de la cámara y dos motores eléctricos de continua (DC) con los actuadores de modulación de anchura de pulsos (PWM. *Pulse Width Modulation*) incorporados en cada rueda.
- **Fuente de energía:** Dos baterías recargables 18650 son las encargadas de suministrar la energía necesaria que alimenta tanto al robot como a la placa procesadora. Tienen un voltaje nominal de 3,7V y su carga eléctrica es de 2400mA/h.
- **Unidad computacional:** Se ha seleccionado una Raspberry Pi 4 con 4GB de memoria RAM, ya que ofrece recursos y rendimiento suficiente tanto para alojar el servidor como para realizar al mismo tiempo el control del Alfabot. Por tanto, al tratarse de una placa computadora reducida, tiene una velocidad de procesador comparable con sistemas de PC. Es capaz de controlar los movimientos y sensores del robot a través de la conexión de éstos a los GPIOs de la Raspberry, igual que de captar los datos que éstos manden, mediante un

código Python creado y alojado en ella. La conexión Wi-Fi de la que dispone hace posible llegar a ella desde cualquier dispositivo conectado a la misma red.

- Conexión de red: Para controlar remotamente el robot mediante la aplicación web diseñada se necesita conexión a Internet. Para ello, es utilizado como puente entre el dispositivo del usuario que desee acceder al sitio web y la unidad de control del robot, un *router*, al que irá conectado mediante Wi-Fi, de igual forma que el usuario.
- Aplicación web: La web es desarrollada mediante Flask, un *framework* de aplicación web de Python. Ésta consta de tres partes: archivo **Python** en el que se importa *Flask*, donde está el programa de la aplicación que realiza el sitio web y a la vez que controla el robot, archivo **HTML**, para que tenga formato web se necesita el lenguaje propio de desarrollo de páginas, y por último, el archivo **CSS**, para darle el diseño deseado, donde estará el estilo de la página. El diseño debe ajustarse a lo necesario para que el usuario pueda acceder e interactuar fácilmente.
- Servidor web: Como Flask se instala en la Raspberry, es ésta misma la que hace de servidor web, donde se aloja la página.

1.5. Proyectos entregables

Para cumplir con el objetivo dicho, se llevarán a cabo los siguientes entregables:

- Memoria: Envuelve toda la documentación relacionada con el desarrollo del proyecto del control del robot.
- Programas de hardware: Implica todos los códigos utilizados para conseguir el control sobre el robot: el dominio de sus movimientos, la comunicación e intercambio de datos con los sensores y la cámara, y la transmisión de datos con la página web.
- Aplicación web: Es el diseño y desarrollo de la página, la cual es accesible a todos los usuarios que busquen interactuar con el robot de forma remota, usando como servidor de ésta, la propia placa computadora.

1.6. Motivación de este trabajo

Desde que curso Bachillerato he tenido un especial interés en la robótica y el espacio, debatiéndome siempre en cuál de los dos me gustaría más. Acabé decantándome por el espacio al elegir mi carrera universitaria. Durante los años que he cursado el grado de Ingeniería en Vehículos Aeroespaciales, mis conocimientos sobre

el espacio han aumentado, pero lamentablemente no ha sido el caso para la tecnología de los robots. He podido cursar algunas asignaturas basadas en la electrónica, que sería lo más cercano, pero no lo deseado ni suficiente, hasta llegar al punto de plantearme realizar un Máster relacionado con la robótica.

Entonces no vi mejor opción que unir mis dos pasiones en una: realizar un trabajo de fin de grado que relacione la robótica con posibles aplicaciones en el espacio, para así, empezar a profundizar más en lo que me encantaría que fuera mi futuro laboral y empezar con el aprendizaje e investigación de ello, aun que sea en un pequeño proyecto que solo podría servir como prototipo de algo a mejorar posteriormente.

1.7. Resumen del contenido

En este primer capítulo se han presentado las pautas que se seguirán a lo largo del proyecto y los objetivos a los que se quieren llegar.

En el Capítulo 2 se abordan los conceptos básicos sobre la robótica móvil así como su relación con el sector aeroespacial. También se analizan y estudian los elementos necesarios que componen un robot, como el Alphasbot, así como el funcionamiento que se requiere para su gestión vía web.

El diseño y desarrollo del robot se explica al detalle en el Capítulo 3, siguiendo las especificaciones y el alcance mencionados anteriormente para cumplir con el objetivo fijado.

En el Capítulo 4 se muestran los resultados obtenidos y el funcionamiento del robot, es decir, como el robot es controlado remotamente mediante cualquier dispositivo electrónico conectado a la misma red, y como manda los datos obtenidos por medio de sus sensores. También se incluye un manual a modo de resultado, que el usuario debe realizar para poner en marcha el sistema.

Finalmente en el Capítulo 5 se exponen las conclusiones obtenidas gracias a este proyecto y qué posibilidades habría en un futuro para mejorar y/o continuar el trabajo expuesto.

Capítulo 2

La robótica y el espacio

2.1. Robótica

La robótica es la rama de la tecnología que estudia el diseño y la construcción de robots, que son estructuras electromecánicas que tienen por objetivo sustituir al ser humano en la realización de tareas repetitivas o que requieran de gran esfuerzo y trabajo por parte del mismo [4]. La idea principal que subyace es que, con el uso de robots, se reducen tiempos de producción y costes monetarios, se optimiza el uso de materia prima, reduce errores humanos y en definitiva, mejora la eficiencia de producción y calidad del producto. Se diferencia de un bot en que está compuesto de mecanismos físicos y tangibles, mientras que el primero está alojado en un software dentro de un sistema virtual ubicado en la Nube [5][6].

En base a las aplicaciones, los robots se clasifican en dos grupos principales: industriales y de servicio [7].

Los robots industriales son un elemento indispensable en la mayoría de los procesos de manufactura y empleados por grandes fábricas en trabajos de fundición, paletización, carga y descarga, montaje, entre otros.

En su mayoría suelen ser brazos con elementos terminales controlados a través de una unidad de proceso (Fig. 2.1). Este tipo de robot se trata de un brazo mecánico, normalmente programable y controlable, con funciones parecidas a las de un brazo humano pero con mayor precisión y rapidez. Las diferentes partes del brazo están conectadas a través de articulaciones que permiten el movimiento rotacional, traslacional o lineal [8].



Figura 2.1: Robot colaborativo de ABB: *IRB 14050 Single-Arm YuMi* [9].

Ejemplos de este tipo de robot pueden encontrarse en el sector de la automoción [10]. Uno de los líderes en robots industriales es ABB, que ofrece una completa gamma a elegir en función de la aplicación a realizar, la carga a soportar y el alcance que precisa [9].

Por otro lado, los robots de servicio no suelen realizar tareas repetitivas, sino que se diseñan para responder con soluciones aisladas a problemas concretos, adaptándose al entorno. Pueden llegar a tener autonomía para moverse y desplazarse, así como utilizarse en entornos peligrosos o donde no es posible el acceso de personas. Por ello, requieren mayor grado de proceso por parte de la unidad de control, y un software adecuado, a la hora de analizar la información del entorno con los sensores para cumplir con su función, además de ser controlados remotamente. Algunos ejemplos son los utilizados en la agricultura, la educación, la medicina, en la investigación tanto en vehículos espaciales como submarinos, junto con otros ámbitos [7].

Un ejemplo de robot de servicio podría ser *Pluto* (Fig. 2.2), un perro-robot autónomo de cuatro patas diseñado para el rescate de personas en túneles profundos, rocosos y sin mapear. Este sería un caso de los mencionados de robots pensados en realizar operaciones de riesgo para los humanos [11], capaz de moverse autónomamente por terrenos irregulares e, incluso, detectar desprendimientos para detenerse, o reconocer patrones de objetos como por ejemplo, un extintor. También es capaz de realizar mapas en 3D en tiempo real que puede enviar remotamente al equipo que se encuentra en superficie.

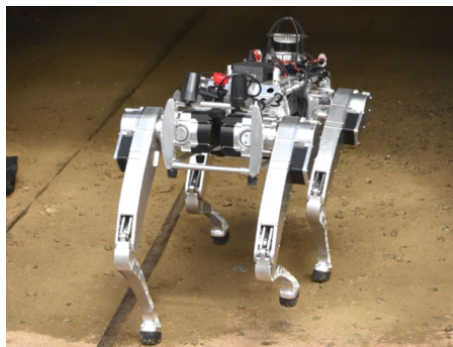


Figura 2.2: Pluto: *Penn Laboratory for Underground Tunnel Operations* [12].

2.2. Robots móviles

A diferencia de los robots industriales que suelen fijarse a la superficie donde trabajan, en los robots de servicio, la plataforma donde van montados dispone de un sistema de locomoción, con ruedas, cadenas o patas, para navegar a través de un determinado ambiente con una cierta autonomía. Por ello, sus aplicaciones son muy variadas y siempre relacionadas con tareas normalmente riesgosas o nocivas, en ámbitos como el de la agricultura, el transporte de cargas peligrosas o tareas de exploración, entre otras.

Aquí, el concepto de autonomía no sólo se refiere al abastecimiento energético, sino también a la capacidad que debe tener el robot de percibir objetos, planificar movimientos y actuar para alcanzar sus objetivos, con poca o prácticamente nula intervención humana [13].

Los robots móviles han de tratar diferentes tipos de incertidumbres sobre la posición o identificación de objetos, ya que para desplazarse a un punto, lo pueden hacer siguiendo diferentes trayectorias, como muestra la Figura 2.3.

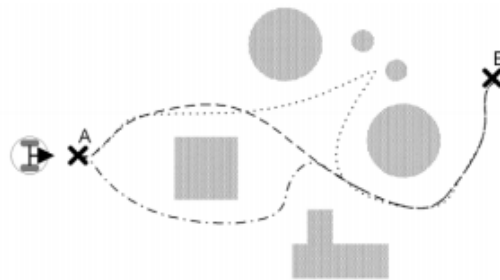


Figura 2.3: Algunas posibles trayectorias a seguir por el robot móvil [13].

2.2.1. Mercado de robots móviles

En los últimos años, los robots móviles autónomos han pasado de ser una idea conceptual a establecerse como un mercado tecnológico estratégico de gran variedad de empresas [14]. El mercado global de robots móviles autónomos ingresó 29.3 mil millones de dólares en 2019, que se espera que alcance 220.6 mil millones en 2030. Se prevé que el mercado tendrá un rápido progreso entre 2020-2030 con una compuesta anual de 18.3% [14]. El rápido desarrollo de la industria electrónica y mecánica es uno de los principales factores que impulsan este crecimiento.

Dentro del mercado de robots móviles es posible distinguir entre productos comerciales, robots a la venta y asequibles a todo aquel que quiera poseer uno, de los que están diseñados específicamente para proyectos de investigación, por ejemplo, y que solo los tienen la empresa que ha solicitado su fabricación. En el primer caso

estaría el Roomba (Fig. 2.5), que es un aspirador doméstico el cual puede comprar cualquier persona a través de su página web [15], y en el segundo, el robot Fedor (Fig. 2.10) diseñado para aplicaciones militares y de exploración.

Por otro lado, están los robots de Boston Dynamics [16], que a pesar de estar a la venta, son robots para tareas muy específicas y con costes elevados que no son comprados para uso personal, sino con fines de investigación comercial, industrial, empresarial y universitaria. Boston Dynamics es un líder mundial en robots móviles, abordando algunos de los desafíos de robótica más difíciles. Se enfoca en crear robots con movilidad avanzada, destreza e inteligencia (Fig. 2.4), por eso la mayoría de sus robots constan de patas y están tan equilibrados como las personas y animales.



Figura 2.4: Spot y Atlas, robots de Boston Dynamics [16].

2.2.2. Aplicaciones de robots móviles

Además de las aplicaciones de ámbito industrial, los robots móviles encuentran aplicación en el ámbito doméstico, del entretenimiento educativo, militar o incluso aeroespacial, entre otros más.

Roomba es un aspirador robótico fabricado por iRobot. Incluye varios sensores que le permiten detectar obstáculos, residuos en el suelo y desniveles como serían unas escaleras. Sus ruedas le permiten realizar giros de 360°. Hasta la fecha actual, 2020, el más inteligente es el Roomba s9+ (Fig. 2.5).

Este robot aspirador tiene conexión a Wi-Fi y con la avanzada tecnología SLAM utiliza un sensor óptico que es capaz de reunir más de 230.400 puntos de datos por segundo para que el procesador de cuatro núcleos a 1,3GHz cree un mapa exacto del lugar (Fig. 2.6). Con la tecnología *Imprint Smart Mapping* controla qué habitaciones limpia y cuándo. Además de esto, viene equipado con hardware capaz de proporcionar funciones inteligentes como aplicaciones con información del robot actualizada,

como muestra la Figura 2.7 [15].



Figura 2.5: Aspirador doméstico: Roomba de iRobot [15].



Figura 2.6: Mapa creado por Roomba [15].

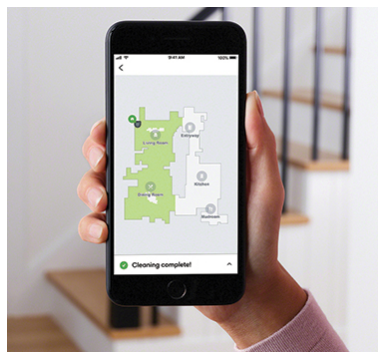


Figura 2.7: Aplicación de iRobot para conocer la actividad del robot [15].

Dentro de la robótica educativa se engloban diferentes formas de aprendizaje y desarrollo de la creatividad: algunas se centran en la construcción, como LEGO Technics, otros en la programación, como Sphero, o un equilibrio entre ambas, como ofrece LEGO Mindstorms [17].

Por ejemplo, se encuentra el primer robot de DJI, la empresa china líder en drones [18], el RoboMaster S1, con el que se busca educar y hacer entender la dinámica de la programación. El RoboMaster S1 (Fig. 2.8) es un robot educativo revolucionario, diseñado para carreras y combates que saca el máximo partido del aprendizaje. Dispone de modos de juego y funciones inteligentes para que los usuarios aumenten su comprensión sobre ciencia, matemáticas, física y programación.



Figura 2.8: Robot educacional: RoboMaster S1 [17].

Sus lenguajes de programación son Scratch 3.0 y Python, que no es tan intuitivo como la interfaz colorida del primero. Consta de Wi-Fi con capacidad de alcance de hasta 130 metros. Es manejable mediante un teléfono o tablet gracias a su app (Fig. 2.9). Un punto débil de este robot es que en tan solo media hora agota toda su batería.



Figura 2.9: Menú principal de la app RoboMaster [17].

Una de las partes más originales del robot S1 son sus ruedas mecanum, capaces de moverse en todas las direcciones gracias a los pequeños rodillos que llevan incorporados con un ángulo de 45° . Cada rueda lleva 12 rodillos inclinados. Otra característica muy atractiva es su inteligencia artificial, ya que posibilita el reconocimiento de personas, gestos y marcadores.

El resultado es un robot muy sensorial capaz de realizar proyectos como identificar y seguir a una persona concreta, transformarse en un vehículo teledirigido accionando el modo 'un jugador', formar una lucha o carrera en el modo batalla lanzando rayos o bolas de gel, entre otros.

Por otro lado, en el ámbito militar, está Fedor, el robot militar explorador ruso.

Estados Unidos, Rusia y China invierten grandes cantidades de dinero en desarrollar robots humanoides que combatan en la Tierra y en el Espacio controlados con Inteligencia Artificial para realizar tareas que pueden ser peligrosas para los humanos [19]. Fedor (Fig. 2.10) es capaz de disparar y conducir vehículos, igual que de participar en misiones espaciales como viajar hasta la Estación Espacial Internacional, a donde fue lanzado en agosto del 2019 [20]. Inicialmente fue diseñado con el fin de participar en situaciones de emergencia de misiones de exploración y rescate.

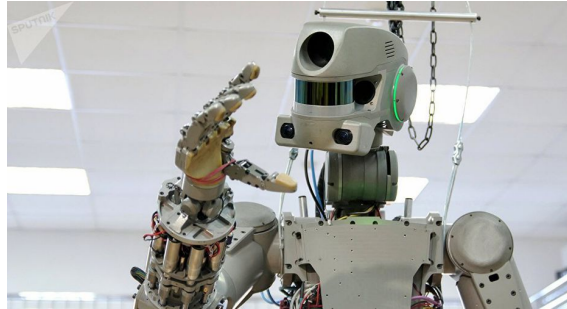


Figura 2.10: Robot militar explorador Fedor [20].

Fedor destaca por su software controlado por Inteligencia Artificial que ha sido entrenado durante varios años por empresas robóticas rusas. Es totalmente autónomo, aun que está preparado para reproducir movimientos teleoperados por humanos en modo avatar. El operario utiliza un dispositivo semejante a los empleados para la Realidad Virtual para ver lo mismo que ve el robot a su alrededor, y mediante unos guantes informa al robot de los movimientos a realizar.

En su viaje a la EEI realizó trabajos de investigación acompañando al resto de cosmonautas de la base [19]. Ha sido preparado para resistir las condiciones climáticas del espacio, como trabajar bajo la ingravidez o radiación.

2.3. Robótica en el ámbito aeroespacial

2.3.1. Introducción

En cuanto al sector aeroespacial, a nivel tecnológico, los esfuerzos se han centrado tanto en el estudio científico de la Tierra, de los planetas y de los astros del espacio. Para ello, son necesarias las misiones espaciales, tripuladas o no, mediante el desarrollo de naves espaciales y cohetes así como satélites espaciales. La exploración espacial empezó con la construcción de los primeros cohetes que permitían poner en órbita satélites artificiales para el estudio de la Tierra como del espacio exterior, y pronto se dio el paso de enviar astronautas al espacio exterior [21].

El primer lanzamiento al espacio fue ocasionado por la Unión Soviética en el 1957, con el satélite artificial Sputnik (Fig. 2.11).

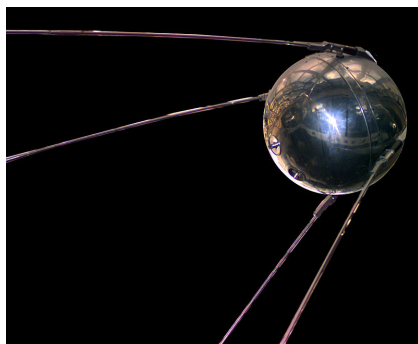


Figura 2.11: La nave Sputnik 1 [22].

Con el primer lanzamiento al espacio de la nave Sputnik, el envío de la perra Laika en el Sputnik II poco después, la llegada del Apolo 11 a la Luna, y el desarrollo de la Estación Espacial Internacional ISS (Fig. 2.12) [23], el campo de la robótica en el espacio fue adquiriendo un empuje en misiones como Viking, Pathfinder o Mars Global Surveyor, entre otras. Aquí, vehículos autónomos fueron equipados con todo tipo de cámaras de vídeo e instrumentos para analizar las características del suelo marciano y entender el planeta.



Figura 2.12: La Estación Espacial Internacional, ISS [24].

Dado que el espacio es un ambiente hostil para el ser humano, las estructuras móviles robotizadas ayudan con servicio tanto a actividades de exploración espacial y planetaria, como al desarrollo de actividades técnicas. Con el desarrollo tecnológico, su complejidad ha ido incrementándose considerablemente llegándose a diferenciar en grupos diferentes a rovers planetarios, de los robots humanoides de apoyo [25].

En comparación con los seres humanos, los robots, espaciales o no, se caracterizan por tener más fuerza para realizar una tarea, velocidad de ejecución, precisión y seguridad. Así como que no necesitan de descanso, alimentación y vestido, todo

y que por contrapartida, funcionan con energía eléctrica. Pero el hecho de poder soportar ambientes hostiles es determinante para usarlos en misiones espaciales.

2.3.2. Rovers

Los rovers son vehículos de exploración diseñados para recorrer superficies de planetas, asteroides o cualquier cuerpo espacial. Empezaron a usarse en misiones tripuladas en la Luna primero (1960), y en Marte después, controlados remotamente.

Algunos ejemplos de Rovers son el Lunar Apollo, entre 1960 y 1970 para misiones lunares de Estados Unidos (EEUU), el Sojourner (1998), Beagle y Opportunity (2004) y Curiosity (2011), éstos últimos enviados a Marte [25].

Para su correcta operación remota, requieren de equipos de control semiautomático que resuelvan el problema del retardo de tiempo debido a la distancia con la Tierra, y de todo el equipamiento para cumplir los objetivos de la misión específica: estudio del suelo, atmósfera, radiación, entre otros. Para ello, se lanzan desde la Tierra al espacio en cápsulas propulsadas por cohetes, que se desprenden al salir de la órbita de la Tierra, e inician su viaje. Cuando la cápsula llega al destino, se coloca en órbita e inspecciona la superficie para buscar el mejor terreno donde posar la sonda. Una parte de la cápsula se desprende y queda orbitando haciendo las funciones de enlace de comunicación entre la Tierra y el Rover, el cual se posa sobre la superficie con el uso de propulsores y paracaídas.

2.3.2.1. Rover Sojourner

El Sojourner (1997) fue un Rover que exploró Marte durante tres meses (Fig. 2.13) que podía moverse a 0,4m/min y que, gracias a sus seis ruedas de 13cm, sorteaba bastantes objetos.

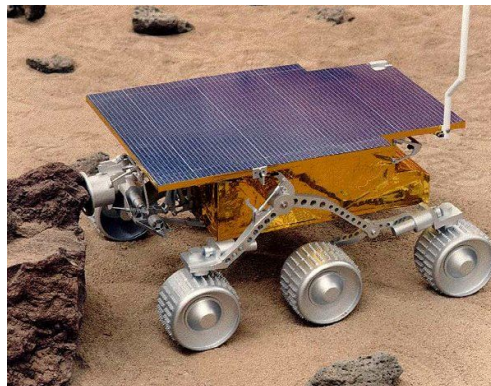


Figura 2.13: Rover Sojourner de la NASA [26].

2.3.2.1.1. Objetivos científicos Los objetivos principales del rover fueron [27]:

- Enviar el rendimiento inicial del vehículo y los datos del experimento tecnológico al módulo de aterrizaje
- Repetir la misma operación en diferentes sitios del planeta
- Adquisición y envío de imágenes
- El desarrollo de procedimientos químicos y análisis APX en todo tipo de suelos y rocas
- Verificación de análisis y envío de datos a la Tierra

2.3.2.1.2. Fuente de energía Los paneles solares de 0,2mm, compuestos por 200 celdas (Fig. 2.14), le daban una autonomía de carga de 16W.

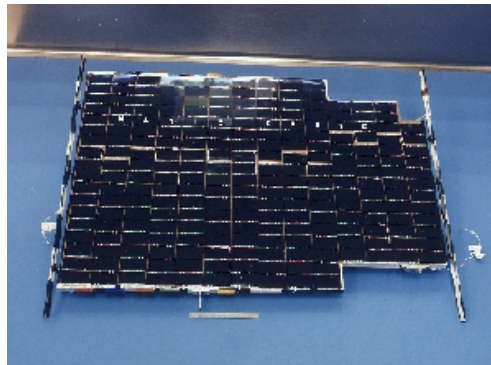


Figura 2.14: Placa solar del Rover Sojourner [28].

Por otro lado, constaba de una batería no recargable para los momentos de poca o ninguna luz, por lo que las operaciones nocturnas eran limitadas. Las baterías utilizadas eran tres de cloruro de litio-tionilo (LiSOCl_2) y D-cell, que le daban una autonomía de 150 W/h a 8-11 V (Fig. 2.15).



Figura 2.15: Baterías del Rover Sojourner [28].

2.3.2.1.3. CPU La unidad de control estaba formada por un Intel 80C85 de 8 bits a 2MHz con cuatro almacenes de memoria: RAM de 64 Kbytes, 16 Kbytes de PROM, 176 Kbytes de almacenamiento no volátil y 512 Kbytes de almacenamiento de datos temporal. Los dispositivos electrónicos que no podían soportar las temperaturas de Marte se alojaban en una caja aislada denominada '*Warm Electronics Box*'.

2.3.2.1.4. Accionamiento El control del movimiento se hacía directamente mediante el encendido o apagado del motor. Un promedio de lecturas del codificador del motor determinaba cuando se apagaban. Una vez apagados, mediante un sistema de láser y cámaras, determinaba la presencia de obstáculos en el camino para variar la trayectoria, y ponerse en marcha de nuevo [26]. Estaba programado para moverse cada 10 segundos, y detenerse otros 20 segundos para observar y entender el terreno que estaba recorriendo.

2.3.2.1.5. Comunicaciones Todas las comunicaciones se realizaban entre el módulo de aterrizaje, la estación base en suelo marciano, y la cápsula en la órbita de Marte. Se realizaban con un módem y una antena UHF, conocida como 'monopolo', que ayudaba a la transmisión efectiva de ondas de radio con la cápsula.

El radiomódem utilizaba RFDC (*Radio Frequency Data Communication*) como método de modulación para transmitir datos de forma inalámbrica a una distancia de hasta decenas de kilómetros, creando redes de radio privadas (PRN) [29] a una frecuencia UHF (450-470 MHz) en el espectro electromagnético dentro del rango de 300 MHz a 3GHz [30], con ráfagas cortas de 2 kilobytes con varios canales de 25 KHz cada uno, y que se usaban para transferir imágenes de la cámara móvil y telemetría que detallaba el estado operativo del rover.

El módem de radio tenía dos circuitos principales separados: uno digital y otro analógico. La placa digital hacía de interfaz entre la parte analógica y la computadora dentro del rover que procesaba los datos, mientras que la placa analógica se encargaba del transmisor UHF y de la modulación bidireccional de señales [31].

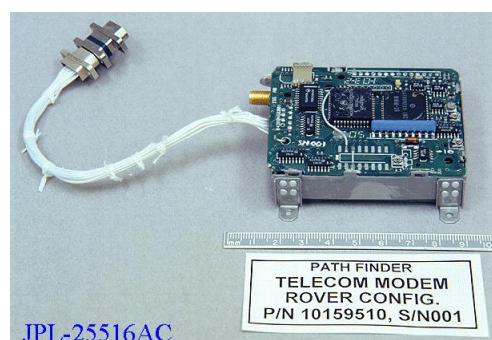


Figura 2.16: Hardware del módem de radio Rover [31].

En condiciones normales de funcionamiento, enviaba periódicamente una señal, denominada 'latido', al módulo de aterrizaje. Si no recibía respuesta, el Rover viajaba nuevamente a la posición donde recibió el último latido.

2.3.2.1.6. Espectrómetros El Espectrómetro de rayos X Alpha Proton (APXS) servía para determinar la composición elemental química de las rocas y polvo de Marte, excepto el hidrógeno (Fig. 2.17).

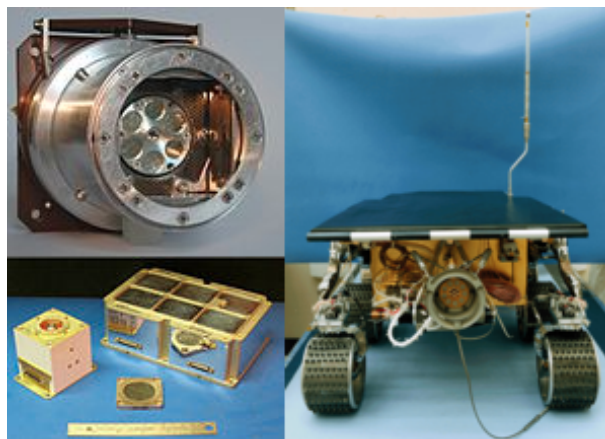


Figura 2.17: Espectrómetro de partículas alfa de rayos X (parte superior izquierda), APXS en la parte posterior de la Mars Pathfinder Sojourner Rover (derecha) [32].

2.3.2.1.7. Cámaras Sojourner constaba de tres cámaras: dos monocromáticas en la parte delantera y una de color en la parte trasera, con gran ancho de banda rojo, verde e infrarrojo. Las dos primeras de 484x768 px proporcionaban imágenes para estereoscopia y mediciones junto con cinco proyectores de banda láser. La trasera proporcionaba información espectral e imágenes para el análisis APXS, capaz de rotar 90° y constaba de los mismos píxeles [27]. Las cámaras se controlaban por la CPU del Rover.

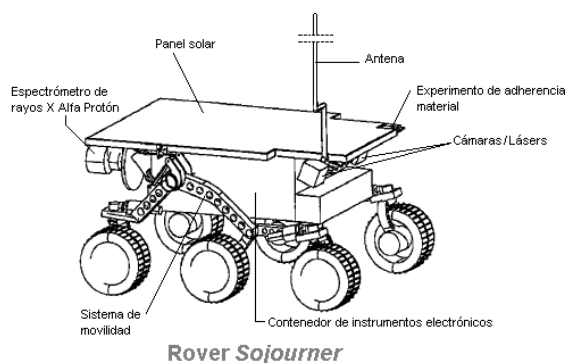


Figura 2.18: Partes e instrumentos del Rover Sojourner [32].

2.3.2.2. Rover Curiosity

Curiosity aterrizó el 6 de agosto de 2012, y aun que la duración prevista era de casi 2 años terrestres, hoy en día aún sigue en operación (Fig. 2.19). Todo y su longitud de 2,7m, es capaz de superar obstáculos de una altura de 75cm. Sus seis ruedas le permiten ir a 90m/h, aún así su velocidad promedio es de 30m/h.



Figura 2.19: Rover Curiosity [33].

2.3.2.2.1. Objetivos científicos El objetivo de su misión consiste en determinar si existió vida en Marte, caracterizar su clima, determinar su geología y recoger información para la exploración humana en Marte. Para ello, realiza las siguientes tareas:

- Determinación de la naturaleza y clasificación de los componentes orgánicos que contienen carbono
- Realización de un inventario de los principales componentes que permiten vida
- Identificación de las características de los procesos biológicos que suceden en Marte

2.3.2.2.2. Fuente de energía El Curiosity funciona con un 'Generador termoeléctrico de radioisótopos multi-misión', MMRTG (Fig. 2.20). Consiste en una cápsula con radioisótopos de plutonio-238 que genera calor y es convertida en electricidad mediante una termopila, llegando a producir 2,5 kW/h al día, mucho más que los paneles solares del Soujourner, que generaban 0,59 kW/h al día. Se estima que tiene una vida útil mínima de 14 años.



Figura 2.20: MMRTG del Curiosity [34].

2.3.2.2.3. CPU Está formada por dos ordenadores idénticos llamados RCE (*Rover Compute Element*), con un sistema operativo en tiempo real (RTOS), *VxWorks*, que procesa los datos al mismo tiempo que se adquieren sin producir retrasos en el búfer [35]. Cada computadora tiene 256 kB de memoria EEPROM, 256 MB de DRAM y 2 GB de memoria flash.

Utiliza la CPU RAD750 como unidad de control, especialmente tratada para resistir condiciones climáticas adversas de alta radiación. Su frecuencia de reloj es de 110 a 200 MHz y puede procesar hasta 400 MIPS (*millions of instructions per second*). La otra computadora es de emergencia y está configurada para funcionar cuando la principal falle.

De los cuatro procesadores de la CPU, uno es un SPARC de 64 bits reservado para controlar los propulsores y los motores en la etapa de descenso a la atmósfera marciana. Otros dos son PowerPC, uno utilizado para llevar a cabo casi todas las operaciones de movimiento en el suelo marciano, y otro gestiona el almacenaje de datos y copias de seguridad. El último es otro SPARC que controla el movimiento del móvil y forma parte de la caja del control del motor [36].

Consta de un sistema de medida inercial IMU (*Inertial Measurement Unit*) que le sirve para medir la fuerza específica de un cuerpo, aceleración, velocidad angular y orientación del Rover, proporcionando información sobre los 3 ejes que le sirven para controlar los movimientos verticales, horizontales y de guiñada.

2.3.2.2.4. Sistema de movilidad Actividades como tomar fotografías, conducir y manejar los instrumentos, se controlan desde el exterior, mandando las órdenes y comandos necesarios al Rover. Esto lo realizan varios operadores que antes de mandar las órdenes, realizan simulaciones [37].

2.3.2.2.5. Comunicaciones El Curiosity transmite directamente información a la Tierra mediante un transmisor y receptor de banda X, de 7 a 11,2 GHz con una velocidad de 32 kbit/s, aunque la ruta de transmisión de datos se hace mediante dos hubs orbitadores existentes en Marte, *Mars Reconnaissance Orbiter* y *2001 Mars Odyssey*, que emiten a más potencia y sus antenas son más grandes que el módulo de aterrizaje, llegando a tasas de comunicación de hasta 2000 kbit/s, el primero, y 256 kbit/s, el segundo. Para comunicarse con ellos se utiliza una radio definida por software, UHF Electra-Lite [38]. Electra (Fig. 2.21) es un paquete de telecomunicaciones que actúa como un satélite, que transmite y amplifica las señales de radio a través de un transpondedor, crea un canal de comunicación entre la fuente transmisora y el receptor. Puede llegar a velocidades de datos de hasta 1 Mbit/s [39].

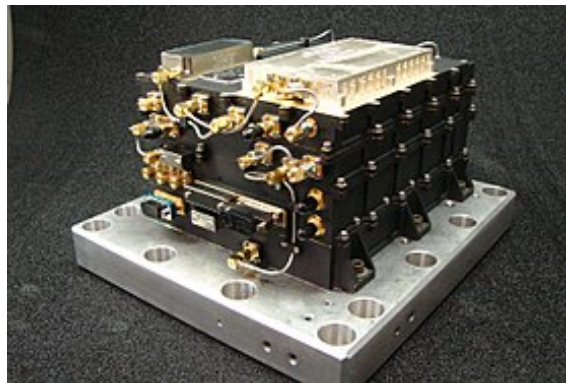


Figura 2.21: Transceptor Electra [39].

Las señales entre la Tierra y Marte (Fig. 2.22) toman una media de 14 minutos y 6 segundos en llegar a su destino.

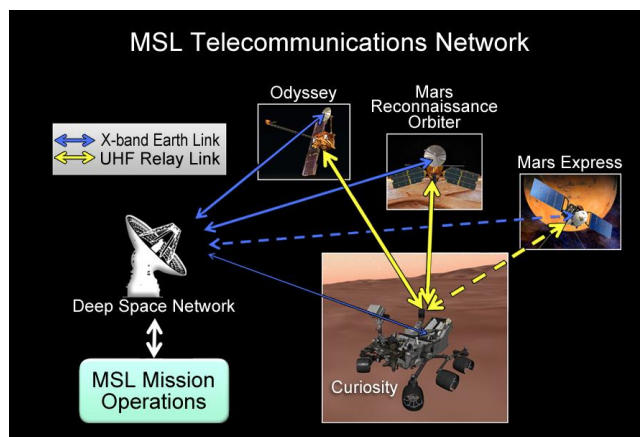


Figura 2.22: Curiosity transmite datos a la Tierra directamente o mediante los satélites en la órbita de Marte [37].

2.3.2.2.6. Espectrómetros Curiosity consta con una versión mejorada del espectrómetro del Sojourner (Fig. 2.17), irradiando muestras con partículas alfa para

luego mapear los espectros de rayos X que se devuelven para determinar la composición química elemental de los cuerpos analizados.

El instrumento consta de una unidad electrónica en el cuerpo del Rover y un cabezal sensor en el brazo robótico que éste contiene.

2.3.2.2.7. Cámaras El Curiosity lleva cámaras incorporadas de alta resolución para buscar características de interés en muestras encontradas. Todas con capacidad de procesamiento de imágenes a bordo. Concretamente consta de 17 cámaras (Fig. 2.23): 8 HazCams, 4 NavCams, 2 MastCams, 1 MAHLI, 1 MARDI y 1 ChemCam [40].

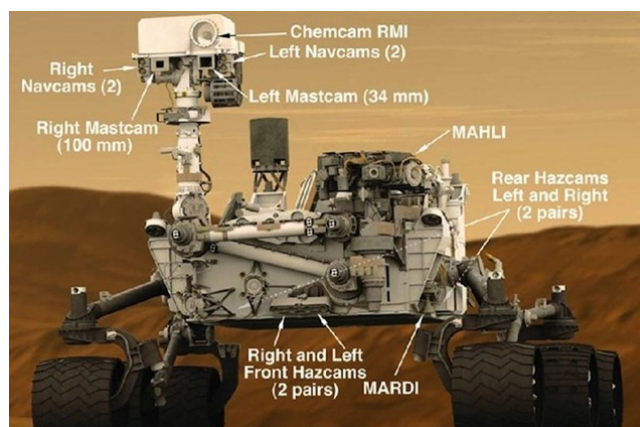


Figura 2.23: Cámaras del Curiosity [41].

2.3.3. Otros robots de apoyo en el espacio

Además de los Rover, existen otras estructuras que se utilizan para dar soporte en el espacio. El brazo manipulador Canadarm fue una estructura montada dentro de los transbordadores espaciales, también denominados '*Shuttle*', de la NASA, y que más adelante se perfeccionó (Canadarm 2) para formar parte permanente de la estación internacional ISS [25].

Por otro lado, Fedor (Fig. 2.10), o Skybot F-850, el robot humanoide que se envió hacia la ISS explicado en la Sección 2.2.2. Tiene capacidades semi-autónomas, con funciones programadas e Inteligencia Artificial.

Se espera que en el futuro, este tipo de robots se controlen de forma remota para sustituir a los astronautas simplificando su trabajo y tareas, hecho que reduciría de manera considerable los costes totales por misión espacial. Los avances en tecnología robótica son cada vez mayores y pueden ser una solución para las empresas aeroespaciales. Se estima que este mercado alcance los 4.400 millones de dólares dentro de la industria en 2023 [42].



Figura 2.24: Canadarm (a la derecha de la imagen) durante la misión del transbordador espacial STS-72 [43].

2.4. Componentes del Alhabot

Al margen de la aplicación específica que pueda desarrollar un robot móvil, además de la estructura (corrocería, ruedas y elementos terminales), existen una serie de componentes que se consideran básicos, como es la unidad de proceso (o CPU), los sensores que sirven tanto para coger información del entorno (temperatura, luz, humedad, etc), como para controlarse a sí mismo (velocidad, posicionamiento, detector de obstáculos, cámara de vídeo, etc) y los actuadores para ejecutar los movimientos, así como interactuar con el entorno. Además, a todo esto, hay que añadir el sistema de alimentación que le suministre la energía eléctrica necesaria para su funcionamiento.

El Alhabot es una plataforma que contiene los elementos mínimos indispensables para desarrollar un robot móvil y controlarlo con una unidad computacional, que puede ser tanto una Raspberry Pi, un Arduino UNO o ambos. Está formada por el chasis, la placa base Alhabot, que contiene la PCB con el hardware y los drivers para conectar tanto la unidad computacional, como los sensores y actuadores necesarios para su movimiento [44]. El hecho de poder conectar ambas unidades, Raspberry Pi y Arduino, permite distribuir las diferentes tareas del robot, por ejemplo, haciendo que el Arduino controle los movimientos: esquivar obstáculos, seguimiento de línea, monitoreo de video, control remoto por infrarrojos, entre otras; y que la Raspberry, con más capacidad de proceso, se dedique al proceso de la información y conectividad con otros dispositivos computacionales.

Las partes que componen la placa base Alhabot se resumen en la Figura 2.25.

- | | |
|-----------------------------|----------------------------------------|
| 1. Conector de Raspberry Pi | 12. Interfaz SPI NRF24l01 para WiFi |
| 2. Conector de Arduino | 13. Interfaz para línea de seguimiento |
| 3. Interfaz motor | 14. Convertidor AD TLC1543 |

- | | |
|---------------------------------------|---------------------------------------|
| 4. Interfaz de ultrasónidos o cámara | 15. Driver de potencia LM298P |
| 5. Interfaz para servo-motores | 16. Diodo anti-retroceso |
| 6. Interfaz de detector de obstáculos | 17. Interruptor ON/OFF |
| 7. Encoder de sensor de velocidad | 18. Regulador de tensión de 5V LM2596 |
| 8. Conector de baterías 18650 | 19. Indicador de potencia |
| 9. Entrada de energía | 20. Interruptor para UART-RS232 |
| 10. Expansión de cabecera Arduino | 21. Receptor de infrarrojos |
| 11. Interfaz UART para Bluetooth | 22. Selección de RPi o Arduino |

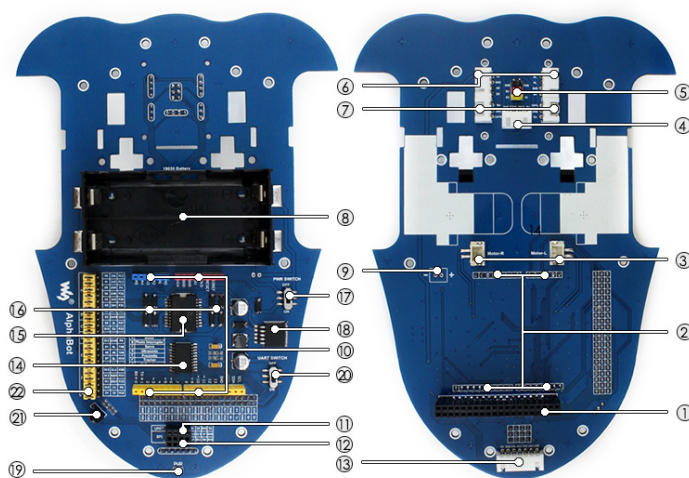


Figura 2.25: Elementos de la placa base del Alfabot [44].

Las características y funcionamiento de los elementos de la placa base son las siguientes:

- Los dos motores que mueven las ruedas están accionados por el driver L298P. Proporciona una corriente máxima de 2A y lleva un diodo de protección para suavizar los picos que se producen durante la transición de la señal de actuación.
- Dos sensores infrarrojos. No tienen tanta precisión como sensores de ultrasónidos, pero cumplen con la función de detectar obstáculos. El sensor de ultrasónidos solo está preparado para trabajar con Arduino en la placa base.
- Los encoders en los motores miden la velocidad y el espacio recorrido en base al conteo de pulsos por unidad de tiempo, que se producen cuando el eje corta el haz de luz (ver sección 2.4.3.1).
- Mediante un control remoto, en forma de mando a distancia, y otro sensor de infrarrojos, se puede controlar los movimientos del Alfabot manualmente.

- El regulador LM2596 sirve para adaptar la alimentación proveniente de las baterías 18650 a la Raspberry Pi o Arduino a 5V.
- Módulo con cinco sensores ópticos analógicos para el seguimiento de línea.
- Conversor AD TLC1543 para que la RPi pueda usar los sensores analógicos, ya que sólo dispone de puertos digitales.
- Brazo robótico donde va implementada una cámara Pi o un sensor Ultrasónico (en el caso del Arduino). Se compone de dos servos que le permiten dotarse de movilidad giratoria.
- Cámara de color de 8 Mpx.

2.4.1. Placa controladora

Todo robot necesita una unidad de control que sea la encargada de analizar la información recibida por los diferentes sensores, procesarla y que, según las órdenes de usuario, accione los actuadores para que los motores lo muevan [45]. Esto se puede hacer de muchas maneras.

En lazo abierto, que equivale a un funcionamiento más manual en el que el usuario envía órdenes mediante un ordenador, y el controlador las ejecuta directamente sobre los actuadores. Y por otro lado, en un proceso más automatizado, el usuario simplemente indica al robot el objetivo, y mediante el programa correspondiente, el robot ejecuta las instrucciones con las secuencias que permite llevarlo a cabo.

Un microcontrolador contiene una unidad central de procesamiento (CPU), encargada de procesar la información, además de la memoria RAM y ROM donde guarda el programa y las variables de proceso, junto con los puertos de entrada y salida, que es donde se conectan los periféricos. Todos estos bloques están interconectados internamente formando una arquitectura flexible que simplifica el diseño de aplicaciones y permite reprogramarlo para realizar diferentes tareas (Fig. 2.26).

A nivel general, los bloques principales de un microcontrolador son los siguientes [46]:

2.4.1.1. Microprocesador

- ALU (*Arithmetic Logic Unit*): Es la unidad principal de cálculo que realiza las operaciones lógicas (solo tiene entradas y salidas con respuesta lógica, 0 o 1), aritméticas (suma, resta, multiplicación y división) y misceláneas (como por ejemplo, transferencia de bits).

- Unidad de control: implementa la máquina de estados con la circuitería necesaria para distribuir la lógica de las señales que permiten la realimentación según el programa cargado.
- Registros: son los elementos que se utilizan para almacenar datos. La memoria RAM es volátil y los datos se pierden cuando se apaga el suministro de energía. Se utiliza para almacenar variables temporales. La memoria FLASH, en cambio, guarda el código del programa que implementa la máquina de estado para la ejecución secuencial de cada línea, mientras que la memoria CACHE guarda los datos imprescindibles para el arranque (*boot*).

2.4.1.2. Periféricos

Corresponden a los circuitos (o hardware) necesarios para que el microcontrolador interactúe con dispositivos externos. Éstos son las salidas digitales y analógicas, interfaces de comunicación RS-232, I2C, SPI, entre otros.

- Puertos de entrada y/o salida paralelos: Permite leer información de elementos externos como pulsadores e interruptores, además de otros sensores digitales. También permite implementar mecanismos de control PWM (*Pulse Width Modulation*) para actuadores. En algunos casos, los sensores pueden utilizar varias de estas líneas y trabajar según la longitud del Bus, que puede ser de 8 bits y llegar a 64 bits, según el microcontrolador.
- Puertos seriales: En estos pines, la información digital (en bytes) fluye por tramas o flancos de bajada/subida de su tensión eléctrica. En comunicaciones asíncronas como el puerto serie UART-RS232 sólo se utilizan dos pines: uno de recepción o lectura de datos (Rx) y otro para el envío (Tx). En comunicaciones síncronas como I2C o SPI, se requieren líneas adicionales para el control de la transmisión entre varios dispositivos.
- Convertidores Analógico-Digital (AD) y Digital-Analógico (DA): El primero sirve para capturar los datos de los sensores cuya salida de tensión se especifica en un rango analógico (i.e. de 0 a 5V), convirtiendo este valor en un código digital cuya profundidad en bits está delimitada por el propio convertidor. El convertidor DA, en cambio, realiza el proceso inverso. Convierte un código digital calculado en el microprocesador a una salida de tensión analógica.

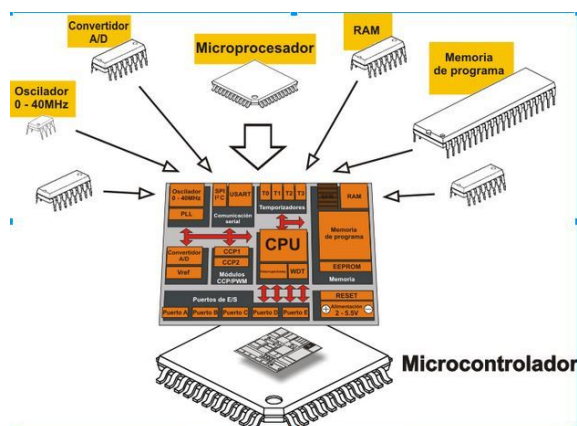


Figura 2.26: Elementos que componen un microcontrolador [47].

2.4.1.3. Raspberry Pi 4B

Esta plataforma (Fig. 2.27) consta de un procesador ARM Cortex-A72 de 1,5GHz de velocidad que funciona con un sistema operativo Raspberry Pi OS. La memoria RAM puede variar según el modelo entre 1, 2, 4 y 8 GB. Incorpora comunicación inalámbrica Bluetooth 5.0, Wi-Fi 802.11ac y Ethernet 8P8C para conectarse a Internet.

A diferencia de los portátiles 'laptops' o PCs, dispone de un puerto externo GPIO (*General Purpose Input Output*), sobre el que se pueden conectar dispositivos, de manera que también permite cargar y correr programas como en un microcontrolador, mediante una tarjeta uSD. Éstos se implementan en lenguaje Python que ya lleva incorporado al instalar el sistema operativo. Además de los 40 pines del puerto GPIO, dispone de dos entradas micro HDMI, y 4 puertos USB, además de dos terminales para conectar cámaras [48].

Los terminales del conector GPIO (Fig. 2.28) soportan componentes de 3,3V y están diseñados para ofrecer 3mA a cada pin [49].



Figura 2.27: Placa Raspberry Pi 4B [48].

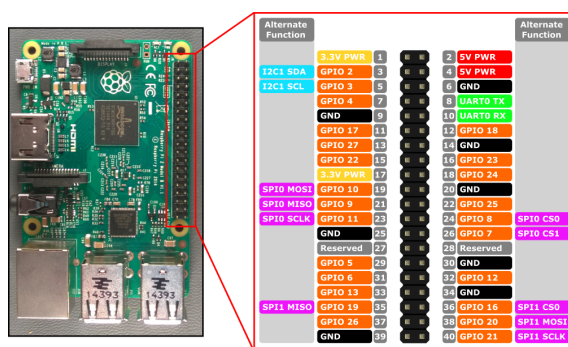


Figura 2.28: GPIO de Raspberry Pi marcados de color naranja [49].

2.4.1.4. Arduino

Arduino UNO (Fig. 2.29) es una placa potente y la más utilizada y documentada de toda la familia Arduino [50]. Es un microcontrolador basado en el ATmega328P, tiene 14 pines digitales de entrada y salida, 6 entradas analógicas, una conexión USB y un USB para su alimentación (Fig. 2.30).

Es de código abierto destinado al desarrollo de aplicaciones IoT para personas con poco conocimiento en electrónica, ya que es fácil de usar tanto en el hardware como en el software. Lee entradas, ya sea en formato analógico o digital, y las convierte en una acción de control (o salida), pudiendo activar un motor, encender un led o publicar algo en línea o escrito en un archivo.



Figura 2.29: Placa Arduino UNO [50].

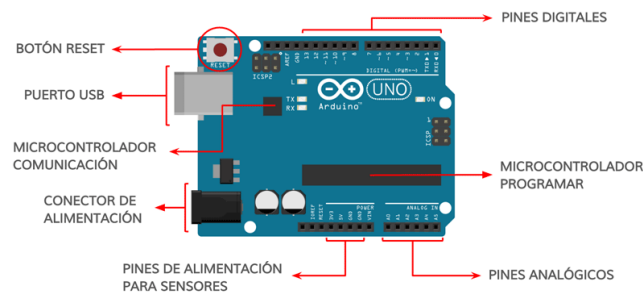


Figura 2.30: Pines y partes del Arduino UNO [51].

2.4.2. Sensores

Los sensores adquieren información del entorno convirtiendo una magnitud física o química en una variable eléctrica. En función del tipo de variable y el principio físico usado para esta transducción, existen multitud de sensores [45] [52]. Entre los que se suelen usar en todo tipos de robots destacan los siguientes [53]:

- Luz
- Sonido
- Detección de objetos (capacitivos, inductivos y ultrasonido)
- Temperatura
- Humedad
- Presión y/o fuerza
- Gravedad
- Velocidad
- Distancia y posición

2.4.2.1. Sensores infrarrojos de detección de obstáculos y seguimiento de líneas

El sensor de obstáculos funciona con un sensor de intensidad lumínica, basado en fotodiodos que responden al cambio de luz para detectar el objeto. Esta luz es detectable sólo en un pequeño intervalo dentro del rango de la radiación electromagnética respecto la longitud de onda, desde los 380 nm a los 780 nm [54].

Para ello, se utiliza un dispositivo emisor que emite luz para que ésta sea detectada por el fotodiodo. En función de la detección de luz y la señal transducida, se decide si el obstáculo ha sido detectado o no. A diferencia de los sensores capacitivos e inductivos, que necesitan que el objeto esté cerca del transductor para ser detectado, los sensores de luz infrarroja ópticos permiten detectar objetos a más distancia [55].

Existen 3 tipos de funcionamiento para los sensores fotoeléctricos [56]:

Modelo de barrera En el detector de barrera (Fig. 2.31), el transmisor y el receptor de luz están separados. Cuando se interpone un objeto entre ellos, se interrumpe la luz y se detecta el obstáculo. En este método, las características físicas del objeto como el color, la textura o el brillo, no influyen en el resultado de la detección del obstáculo ya que la luz ha de llegar al receptor para obtener el sensado, por lo que es bastante fiable y consigue distancias más largas de detección. Tiene el inconveniente de que el emisor y receptor se han de montar en dos sitios diferentes, por lo que raramente se suele utilizar en robots móviles.

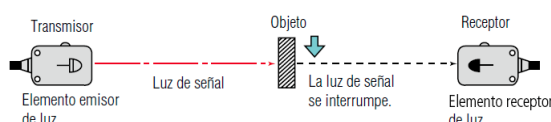


Figura 2.31: Sensor fotoeléctrico tipo barrera [56].

Modelo reflectivo En el modelo reflectivo, en cambio, transmisor y receptor forman parte de un solo cuerpo (Fig. 2.32). En este caso, el haz de luz infrarroja rebota contra el objeto y la recibe el receptor. Es de los más económicos, pero tiene por contrapartida que la longitud de onda de la luz del emisor debe ser compatible con el objeto a detectar para que la luz rebote. Concretamente, el objeto no puede ser de colores oscuros con poco factor de reflexión [57].

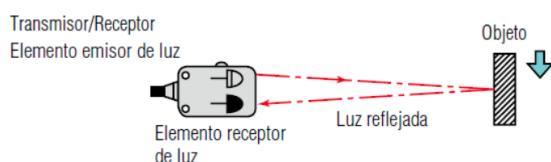


Figura 2.32: Sensor fotoeléctrico tipo reflectivo [56].

Modelo retroreflectivo Este principio (Fig. 2.33) es parecido al anterior, pero utiliza un reflector catadióptrico externo para reflejar la luz. Tiene la ventaja de que no depende del color del objeto pero al igual que el sensor de barrera, no se utiliza en robots móviles.

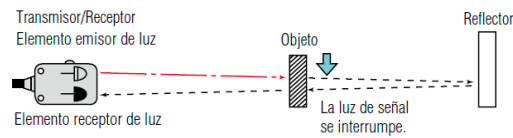


Figura 2.33: Sensor fotoeléctrico tipo retroreflectivo [56].

■ FC-51

El Alphabot utiliza el módulo FC-51 (Fig's 2.34 y 2.35) que consiste en un emisor-receptor de luz infrarroja y un comparador de tensión. El transmisor es un LED infrarrojo que funciona a partir de 3V DC y tiene un consumo de 20 mA. El comparador LM393 recibe y controla la señal analógica que se convierte a digital. El potenciómetro forma un divisor de tensión e implementa la referencia de comparación de tensión para detectar el objeto. Ajustándolo convenientemente, permite cambiar la sensibilidad de detección del sensor, encendiendo el LED verde cuando el objeto es detectado.

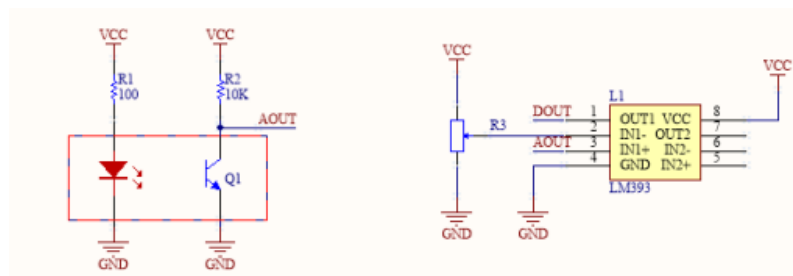


Figura 2.34: Circuito acondicionador de detector óptico: Conexiónado y esquema eléctrico [44].

El Alphabot dispone de dos módulos detectores de este tipo: uno en la parte izquierda y otro en la derecha, cuyas salidas DOUT del comparador LM393 se conectan al terminal GPIO, pines 19 y 16 respectivamente.

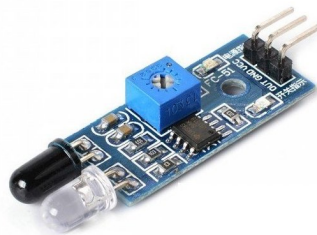


Figura 2.35: Sensor FC-51 [58].

- ITR20001/T

Ilustrado en la Figura 2.36. Su funcionamiento es similar al FC-51. El receptor tiene un sensor de reflexión de infrarrojos ITR20001/T. Un led emite luz infrarroja continuamente y cuando es reflejada por un obstáculo, la recibe el receptor. La salida del sensor es analógica y es sensible al color y distancia del objeto detectado.



Figura 2.36: Módulo siguelíneas [44].

Consta de 5 canales de sensores (Fig. 2.37) para obtener la posición de la línea oscura que se encuentra en el suelo. Es un módulo de seguimiento de línea. Los 5 sensores están conectados a un conversor analógico digital TLC1543.

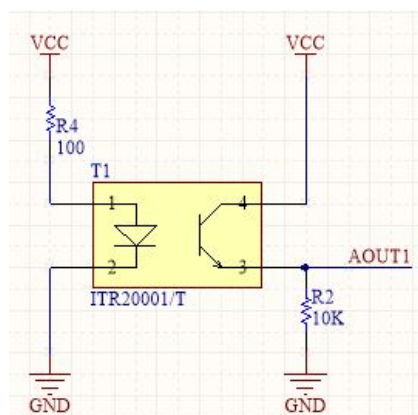


Figura 2.37: Conexión eléctrica de los sensores IR [44].

2.4.2.2. Detección por ultrasonidos

Mediante un sensor que emite ultrasonidos se detectan los objetos escuchando las ondas que rebotan en ellos (Fig. 2.38). Con este principio, también es posible medir la distancia a la que se encuentra el objeto aunque la precisión de medida sea algo limitada.

El principio de funcionamiento consiste en lanzar un tren de pulsos ultrasónicos y medir el tiempo transcurrido entre la emisión y el retorno para calcular la distancia.

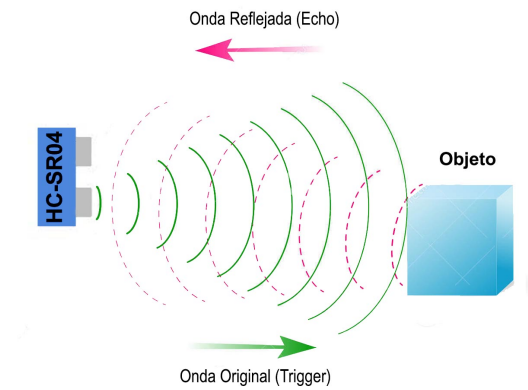


Figura 2.38: Imagen que ilustra cómo viajan las ondas de un sensor ultrasónico [59].

El Alfabot puede incorporar este sensor, además de los dos sensores infrarrojos, para medir la distancia a los obstáculos, siempre que tenga como CPU el Arduino UNO. En este caso, hay que usar el módulo de ultrasonidos HC-SR04 (Fig. 2.39).

Éste dispone de cuatro terminales: alimentación (VCC y GND), activación de transmisión (TRIG) y recepción (ECHO). Para medir la distancia al objeto hay que medir el ancho de pulso recibido por el terminal ECHO (Fig. 2.40). Para ello, hay que enviar un pulso de $10\ \mu s$ por el pin TRIG para que inicie la medición. Entonces, el HC-SR04 transmite 8 series de pulsos a 40kHz, para, y espera la recepción de las señales rebotadas. Si detecta el objeto, el terminal ECHO generará un pulso cuyo ancho será proporcional a la distancia del objeto. La distancia se calcula como $d = \frac{1}{58} \cdot t_{width}$, donde t_{width} es el ancho del pulso en este pin.



Figura 2.39: Sensor HC-SR04 [60].

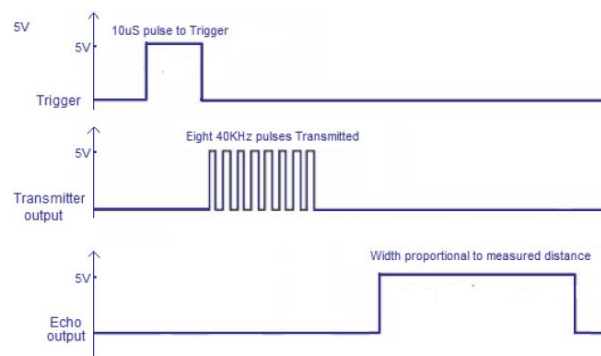


Figura 2.40: Pulsos cuadrados transmitidos [44].

2.4.2.3. Sensores de temperatura

Un sensor de temperatura detecta las variaciones en la temperatura del aire o del agua y las transforma en una señal eléctrica enviándola a un sistema electrónico. Normalmente suele estar formado por el elemento sensor, la cáscara que lo envuelve que está rellena de un material muy conductor, el cual es capaz de variar la resistencia eléctrica en función de la temperatura ambiente, y del cable al que se conectarán al equipo electrónico [61].

Hay tres tipos de sensores de temperatura [61]:

Termistores Un termistor está basado principalmente en que el comportamiento de la resistencia de materiales semiconductores es variable en función de la temperatura.

RTD Es un detector de temperatura resistivo, se basa en la variación de la resistencia de un conductor con la temperatura.

Termopares Recibe este nombre por estar formado por dos metales. Su funcionamiento está basado en el efecto termoeléctrico, que permite transformar directamente el calor en electricidad. Genera una tensión en función de la temperatura que se le aplica al sensor.

2.4.2.4. Sensores de humedad

Algunos sensores de temperatura se combinan con sensores de humedad, dispositivos que miden la humedad relativa de un área específica. Pueden ser analógicos o digitales [62].

Analógicos Miden la humedad relativa del ambiente mediante un condensador. El sensor está hecho de vidrio o cerámica, mientras que el material aislante está formado por polímero, el cual absorbe y libera el agua basándose en la humedad relativa.

Digitales Un sensor digital de humedad funciona a través de dos microsensors que se calibran a la humedad relativa del ambiente. Se convierten en formato digital a través de un proceso realizado mediante un chip situado en el mismo circuito.

- DHT11

Mostrado en la Figura 2.42. Se compone de un sensor capacitivo, para medir la humedad, y de un termistor. Tiene una alimentación de 3,5-5V a 2,5mA con señal de salida digital, por ello gozan de alta fiabilidad y estabilidad. Consta de un microcontrolador pequeño para tratar la señal.

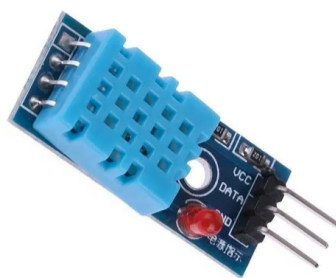


Figura 2.41: Sensor DHT11 [63].

Aunque la salida sea digital, se trata de un dispositivo analógico que se convierte en digital y se envía al microcontrolador. La trama de datos es de 40 bits correspondiente a la información de humedad y temperatura del DHT11 (Fig. 2.42). El primer grupo de 8 bits es la parte entera de la humedad y el segundo grupo la parte decimal. Lo mismo pasa con el tercer y cuarto grupo, la parte entera y decimal de la temperatura. Por último los bits de paridad para confirmar que no hay datos corruptos. Para ello, suman los 4 primeros grupos de 8 bits, donde el resultado debe ser igual a los bits de paridad.

| | | | | |
|----------------|----------------|--------------------|--------------------|-----------------|
| 0011 0101 | 0000 0000 | 0001 1000 | 0000 0000 | 0100 1001 |
| 8 bits humedad | 8 bits humedad | 8 bits temperatura | 8 bits temperatura | bits de paridad |

Figura 2.42: Trama de datos de bits del DHT11 [63].

2.4.2.5. Sensores de velocidad

Los sensores de velocidad tienen por objetivo obtener la velocidad de movimiento a partir de la frecuencia de giro de las ruedas. En el caso del Alfabot, esto se hace con el módulo WYC-H206 (Fig. 2.43), un optointerruptor parecido al detector de obstáculos que funciona como encoder incremental y que cuenta el número de pulsos que se producen por unidad de tiempo.

Consta de una rueda estriada acoplada a un eje giratorio (Fig. 2.44). Cuando éste gira, las estrías rompen la fuente de luz emisora infrarroja que es interpretada por el fototransistor como un tren de pulsos digitales. Interpretando esta señal en el tiempo, es posible obtener y controlar distintos parámetros de movimiento: como la dirección del movimiento, el sentido de giro, la velocidad, etc [64].

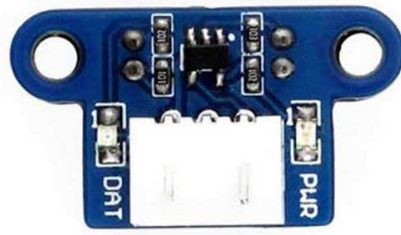


Figura 2.43: Sensor WYC-H206 [44].



Figura 2.44: Encoder del Alhabot [44].

2.4.3. Actuadores eléctricos

Los actuadores son los elementos eléctricos que se encargan de accionar los motores para el movimiento del robot. Para ello, se tienen que transformar las señales eléctricas generadas por la unidad de control, en otras de más potencia que permitan mover los ejes de las ruedas.

2.4.3.1. Motores DC y Servomotores

- Motores de corriente continua: transforman la energía eléctrica en mecánica (de giro) mediante campos magnéticos creados por las dos partes que componen el motor, un estátor y un rotor. El estátor contiene imanes permanentes, y el rotor es el que gira debido a la misma fuerza que aparece por la alimentación de la corriente eléctrica, proveniente del exterior, y el campo magnético de excitación.
- Motores de paso a paso: convierten una serie de impulsos eléctricos en desplazamientos angulares, por lo que son capaces de girar una cantidad de grados. Son ideales para movimientos muy precisos [65].

El Alhabot dispone de dos motores DC (Fig. 2.45), uno para cada rueda, que hay que accionar mediante un driver de corriente. El voltaje de funcionamiento

de los motores es de entre 3 y 6 V. Habitualmente, se usa el módulo L298P, que dispone de un puente en H para mover el eje de las ruedas en ambos sentidos de giro, bloquearlas o liberarlas.

El módulo controlador de motores L298P H-bridge permite controlar la velocidad y la dirección de dos motores de corriente continua o un motor paso a paso gracias a los 2 H-bridge que monta. El rango de tensiones en el que trabaja este módulo va desde 3V hasta 35V, y una intensidad de hasta 2A.

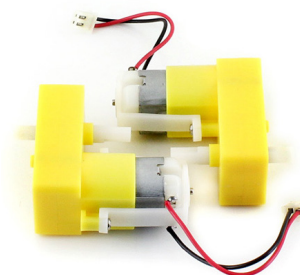


Figura 2.45: Motores DC del AlphaBot [44].

Para invertir el sentido de rotación del motor es necesario invertir la polaridad de la tensión en bornes, para ello está diseñado el circuito llamado puente en H.

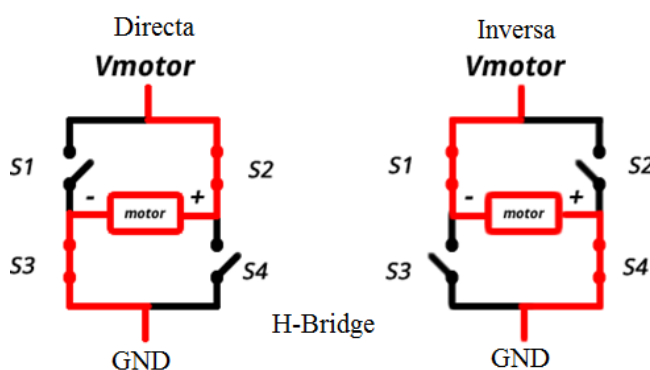


Figura 2.46: Interruptores en puente en H [66].

Para entender el funcionamiento del puente en H, en la Figura 2.46 se puede observar que al usar los interruptores en la posición de la imagen izquierda, el motor gira en sentido directo. Pero si se colocan en la posición de la derecha, girará en sentido contrario, porque la polaridad de la tensión en las entradas del motor ha sido invertida, y por tanto, el sentido de giro.

Otro tipo de motor que también usa el Alphabot es el servomotor, concretamente para mover la posición de la cámara. Éste es un motor que incluye un sistema de autocontrol en lazo cerrado que a partir de la posición y velocidad, con un conjunto

de engranajes reductores, el circuito interno corrige la diferencia llevando la posición al punto exacto establecido por la consigna. Cuenta con tres líneas de control: las dos de alimentación positiva de 5V y GND (naranja y marrón), y la roja de señal (IO), que se utiliza para especificar la consigna de posición [67].



Figura 2.47: Servomotor SG90 [67].

La señal de control IO emite un pulso alto durante un intervalo mínimo de 20ms. Según su duración en estado alto, se traduce en un ángulo de rotación del servo (Tabla 2.1). En la Figura 2.48 se ilustra su funcionamiento.

| Ancho de pulso | Ángulo de rotación |
|----------------|--------------------|
| 0,5ms | 0 grados |
| 1,0 ms | 45 grados |
| 1,5 ms | 90 grados |
| 2,0 ms | 135 grados |
| 2,5 ms | 180 grados |

Tabla 2.1: Control teórico de los motores del AlphaBot.

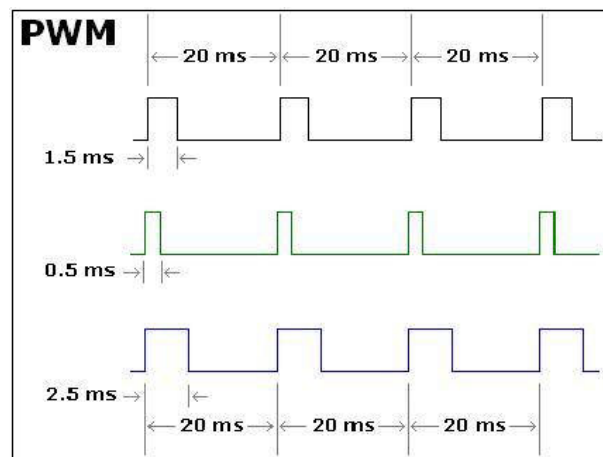


Figura 2.48: Pulsos PWM del servomotor [68].

2.4.4. Otros componentes del AlphaBot

La alimentación de todo el conjunto Alphabot-CPU se basa en dos series de baterías recargables 18650 de Li-Ion, con un voltaje nominal de 3,7V (Fig. 2.49). Este voltaje se regula con el LM2596 y se envía a la unidad de control y los componentes del robot. Su autonomía es de 3 horas aproximadamente.



Figura 2.49: Baterías de Li-Ion 18650 [44].

La placa de cámara Raspberry Pi (Fig. 2.50) con enfoque fijo de 8 megapíxeles de alta definición HD, utiliza el sensor de imagen IMX219PQ de Sony que ofrece imágenes de vídeo de alta velocidad y sensibilidad. La conexión del módulo de cámara a RPi se hace mediante un cable plano de 15 contactos fijado a las ranuras del módulo en el puerto de interfaz serie de la cámara Pi.



Figura 2.50: Cámara Raspberry Pi para el Alphabot [44].

2.4.5. IoT y desarrollo de aplicaciones remotas

Además de los puertos I/O digitales, otra manera de interaccionar con el microcontrolador mediante dispositivos externos consiste en utilizar protocolos M2M (*Machine-to-Machine*). Hoy en día, existen gran variedad de protocolos M2M desde la comunicación serie cableada como RS232, I2C o SPI; las comunicaciones industriales como DeviceNet, Ethernet, IP, y Modbus; hasta comunicaciones inalámbricas: ZigBie, Bluetooth, Wi-Fi o LoRa, entre otras.

Al mismo tiempo, se ha extendido el concepto de *Internet of Things*, IoT, una concepción que se basa en que los objetos, con pequeñas unidades computacionales y sensores, tienen conexión a Internet, lo que les permite enviar y recibir datos sin intervención humana [69]. Antiguamente, el M2M se limitaba a la conexión entre dos máquinas de cualquier medio, mientras que IoT se centraba en las comunicaciones inalámbricas, mayoritariamente 3G y 4G. Pero actualmente, existen protocolos M2M como del LCM [70], que pueden llegar a formar una red LAN o WAN para comunicaciones punto-a-punto entre sensores, máquinas y controles.

Entre los 7 medios de comunicación inalámbrica se encuentran [69]:

- **Satélite:** Inicialmente comprendía el uso para telefonía móvil y mensajería (2G y 3G), pero actualmente el 4G incluye acceso a Internet. Se espera que el 5G incluya protocolos M2M para aplicaciones IoT. Aquí, la comunicación de datos no es directa, sino que tiene que pasar por satélite.
- **Wi-Fi:** Se suele utilizar mayoritariamente para acceso inalámbrico a Internet a nivel local (WLAN). Dado que trabaja con tasas de comunicación elevadas, suele tener un consumo energético elevado.
- **Radio Frecuencia (RF):** Es la forma más fácil de comunicación entre dispositivos que usan la banda de UHF (300MHz a 3GHz). En este rango, se incluyen sistemas como ZigBee (2,4GHz), LoRa/LoRaWAN o SigFox (868MHz en Europa) o RF de 409MHz. Dado que están pensados para redes con dispositivos de poco flujo de datos, su consumo es considerablemente más bajo en comparación con Wi-Fi.
- **RFID:** Es el uso inalámbrico de campos electromagnéticos para identificar etiquetas de objetos mediante un lector activo.
- **Bluetooth:** Posibilita el intercambio de gran cantidad de datos entre dispositivos a distancias cortas utilizando ondas cortas de radio UHF. Tiene una tasa de velocidad comparable al Wi-Fi, pero no se usa para Internet.
- **NFC (comunicación de campo cercano):** Utiliza la inducción electromagnética entre dos antenas cercanas formando un transformador de núcleo de aire. Hay un iniciador y una tarjeta: el iniciador genera un campo de RF que puede alimentar una tarjeta. Se usa en aplicaciones de pago sin contacto, tiquets electrónicos y pago por móvil.

Para desarrollar una plataforma que permita a un usuario interactuar con los controles del Alphabot de manera remota, existen dos opciones principales: desarrollar una interfaz gráfica o alojar una página web HTTP en la propia unidad de control del Alphabot a la que se pueda acceder remotamente.

De estas dos opciones, la más versátil y flexible, desde el punto de vista de un usuario, es la segunda opción, ya que no depende del sistema operativo OS

(Windows, Linux o MAC) ni del tipo de ordenador (portátil o PC) por el que un usuario externo acceda a ella. El usuario, simplemente escribe la dirección web (IP) en la barra del navegador y accede normalmente.

El primer caso, en cambio, corresponde a una comunicación M2M entre dos máquinas. Si el programa se aloja en la unidad de control del Alphabot, el usuario necesita acceder (por ejemplo, mediante SSH) para abrir el programa. Por otro lado, si se instala el programa con la interfaz en remoto, éste requiere disponer de todas las operaciones específicas para establecer conexión con el Alphabot. En ambos casos, se requieren de dos códigos, uno en remoto y otro en el Alphabot para desarrollar la aplicación. Mientras que con la segunda opción, todos los programas pueden estar alojados en el servidor del Alphabot. Por lo que se decide diseñar un servidor web.

Un servidor web es una plataforma hardware para albergar aplicaciones web que dan servicios externalizados a usuarios (también denominados clientes) que se conectan a él e intercambian información de manera bidireccional. El diálogo cliente-servidor se realiza a través de un navegador web que, junto con el software alojado en el servidor, permite dicho trasvase de información.

2.4.5.1. Arquitectura servidores web

Un servidor web opera mediante el protocolo HTTP. Se ejecuta manteniéndose a la espera de peticiones de parte del cliente. Las peticiones al servidor se realizan utilizando el método de petición GET a través de la URL del servidor web.

El segundo tipo de petición HTTP más utilizado es POST, el cual envía al servidor datos que se incluyen en el mismo cuerpo del navegador.

El entorno de usuario es el conjunto de elementos del navegador que permiten realizar peticiones. Los elementos más comunes de este entorno son los siguientes:

- **Hipervínculo (enlace o link):** es una porción del contenido web, puede ser texto, imagen y otros elementos que enlaza a una dirección web. Al pulsarlo se genera una petición GET automática a la URL del mismo enlace.
- **Formulario web:** es el envío de los datos de un formulario mediante una petición GET o POST al servidor.
- **Barra de direcciones:** mediante ella se puede acceder manualmente a cualquier dirección URL. Para ello se generará una petición GET desde cualquier navegador.
- **Scripts:** son aplicaciones JavaScript capaces de acceder al estado del navegador y modificar sus datos.

Para el desarrollo de aplicaciones web en servidor se pueden utilizar diferentes herramientas. Las que son objeto de consideración para este trabajo son las siguientes:

- LAMP es el acrónimo de Linux Apache MySQL y PHP (Fig. 2.51). Es un entorno gratuito que se puede instalar tanto en modo local (localhost), como servidor remoto, que funciona bajo sistema operativo Linux. Es capaz de ejecutar páginas web HTML dinámicas. Apache hace referencia al paquete que implementa el servidor, el cual es libre y de código abierto. Es uno de los más utilizados con el 29.1 % de las webs a nivel mundial [71]. MySQL es un gestor de bases de datos que utiliza SQL (*Software Query Language*) como lenguaje para consulta y operación con tablas estructuradas. Por último, PHP actúa como lenguaje de programación de páginas web dinámicas del lado del servidor. A veces, este último puede ser reemplazado por otros lenguajes de programación como Perl o Python, entre otros.

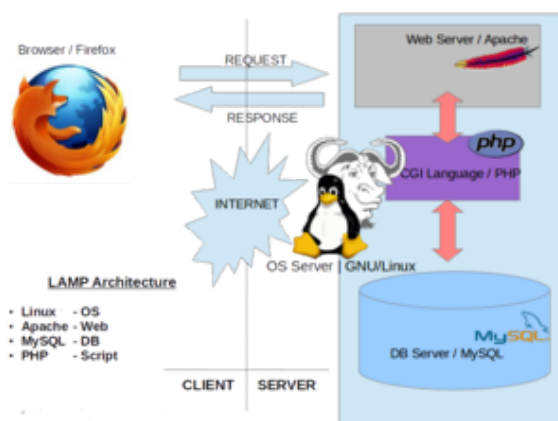


Figura 2.51: Descripción de la conexión entre componentes de LAMP, usando Firefox como ejemplo de navegador [72].

- Node.JS es una multiplataforma de código abierto que permite la creación de servidores web ejecutando JavaScript, su único lenguaje de programación, fuera de un navegador web. Es compatible con distintos sistemas operativos como Linux, MacOS y Microsoft Windows. Trabaja como un bucle de eventos de subproceso único, lo que significa que puede ejecutar tareas paralelas en un mismo hilo.
- Django es un *framework* de código abierto y gratuito basado en Python. Un *framework* web es un conjunto de herramientas que ayudan al desarrollo de sitios web, es decir, a crear su contenido (Fig. 2.52). Las aplicaciones web Django agrupan el código gestionado en ficheros separados:

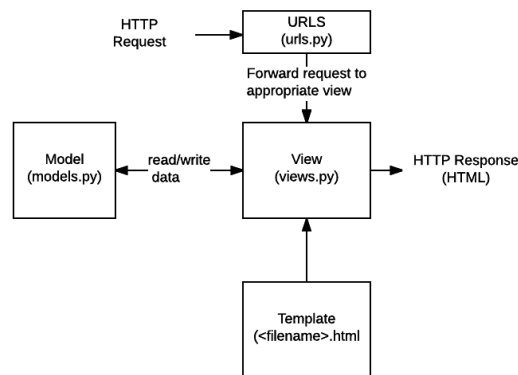


Figura 2.52: Funcionamiento de las solicitudes web con Django [73].

Sigue un patrón MVC (Modelo Servidor-Vista HTML-Controlador/gestor de peticiones). Cuando la página web realiza una petición HTTP al servidor, ésta llega al fichero de Python 'urls.py', que representa un mapeador URL usado para redirigir las peticiones a la vista apropiada, 'views.py'. Una vista es una función que recibe peticiones HTTP y devuelve respuestas HTTP accediendo a los datos necesarios para satisfacer las peticiones mediante modelos (models.py) usando las plantillas (name.html), que son las que definen la estructura de la representación de la respuesta.

- Flask: es un *microframework* escrito en Python que permite crear aplicaciones web de manera sencilla y con un mínimo número de líneas de código. Se denomina micro por el hecho de que solo consta de las herramientas necesarias para crear una web funcional, si se necesitara de más, se pueden instalar una gran variedad de extensiones (*plugins*) que le dan más funcionalidad. También sigue un patrón MVC.

Python es un lenguaje de programación multiparadigma, dinámico y multiplataforma. Esto significa que puede adoptar un estilo particular de programación a cada programador, es decir, permite varios estilos: programación orientada a objetos (los objetos manipulan los datos de entrada para obtener datos de salida específicos), programación imperativa (dar órdenes al ordenador de como realizar acciones en forma de algoritmos) y programación funcional (basado en la programación de funciones matemáticas) [74].

2.5. Discusión

Después de la descripción hecha en las secciones anteriores sobre los posibles componentes de un robot, en este apartado se discuten los pros y contras que llevan a su selección.

Considerando el alcance de este proyecto, se ha previsto para el intercambio de datos entre la aplicación web, la unidad de control y el robot, usar un router doméstico a modo de *gateway* que permita el acceso al robot vía dirección IP en modo local, ya que los objetivos se centran más en el diseño y funcionamiento de la interfaz web en un entorno local (o privado) junto con el control manual remoto del robot, más que en el desarrollo de una aplicación avanzada de acceso público. Por lo que las decisiones más importantes son las relacionadas con los elementos de comunicación entre el Alphabot y la aplicación web, y el software para desarrollar la aplicación web y el servidor.

2.5.1. Controlador del robot

Tal y como se ha visto en el apartado 2.4, el Alphabot acepta el uso tanto de la Raspberry Pi como del Arduino. La Tabla 2.2 resume las características principales de ambos. Sin embargo, por cuestiones prácticas se ha optado por usar exclusivamente la Raspberry Pi como CPU del AlphaBot, que contendrá el servidor con la aplicación web y al mismo tiempo se encargará del control de movimientos del Alphabot.

| CPU | Ventajas y desventajas para el proyecto |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Raspberry Pi | Claramente, tiene gran capacidad de proceso y es equiparable a un ordenador de escritorio, tipo PC. Permite la ejecución de múltiples programas simultáneos, lo que la hace idónea para alojar la web en el mismo controlador del robot, todo y que requiere más consumo y el coste económico es superior comparado con Arduino. |
| Arduino | Es fácil de utilizar y está más preparado para tareas simples repetitivas a nivel hardware, como pueden ser el control de actuadores que accionan los motores del Alphabot, detección de obstáculos, entre otros. Sin embargo, tiene una capacidad de proceso limitada a tareas simples no simultáneas, teniendo que recurrir al uso de interrupciones para gestionar diferentes tareas a la vez. Además, para disponer de comunicación inalámbrica y acceso a red de Internet, requiere del acoplo de otros elementos a modo de Shield, como el ESP8266, para disponer de esta función. |

Tabla 2.2: Selección de la unidad de control para el proyecto.

2.5.2. Comunicación con el *gateway*

Por otro lado, el acceso de usuarios a la interfaz se realiza inalámbricamente a través de un *router* Wi-Fi doméstico en modo local. Esta comunicación *gateway*-servidor es necesaria para la monitorización del estado del robot así como el control de movimientos de manera remota.

Dado que la Raspberry Pi dispone de conexión Wi-Fi y Bluetooth, es posible acceder a la programación del robot, junto con el servidor web, inalámbricamente mediante línea de comandos SSH (*Secure Shell*) a través de dirección IP.

2.5.3. Servidor web

Ya que la programación en Raspberry Pi se realiza mayoritariamente en Python, para el desarrollo de la interficie servidor, se ha buscado una plataforma que trabaje en este lenguaje y que no cargue de sobremanera la CPU del Alfabot. Para ello, existen diferentes opciones (Tabla 2.3).

| Servidor web | Ventajas y desventajas para el proyecto |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LAMP | Suele ser la opción más utilizada para desarrollo de aplicaciones web en data-center, ya que contiene un gran número de paquetes como bases de datos SQL, codificación de archivos por defecto, entre otros, que requieren de tiempo para familiarizarse con ellos y que no son necesarios para este proyecto. No es capaz de transferir datos con scripts Python externos de manera directa, por lo que se han de usar herramientas de terceros, como AJAX, para el intercambio de datos entre la web y el código Python de control del robot. |
| Node.JS | Su lenguaje es JavaScript, pero es capaz de leer y escribir archivos de cualquier tipo, por lo que soluciona el inconveniente de LAMP. No tiene bloqueo de solicitudes ni peticiones, por lo que puede realizar varias interconexiones y ejecutar variedad de tareas a la vez. |

| | |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Django | Está íntegramente desarrollado en Python, dispone de multitud de bibliotecas y es sencillo de usar. Su funcionamiento se basa en LAMP e incorpora multitud de funciones que no son necesarias y cargarían considerablemente la CPU del robot, todo y que el hecho de venir con interfaces prediseñadas, la convierten en una opción atractiva para crear aplicaciones web rápidas y escalables. |
| Flask | También es un entorno para diseño web íntegramente desarrollado en Python. A diferencia de Django, permite el desarrollo de interfaces de servidor sencillas y minimalistas incorporando sólo los módulos necesarios para cumplir con todas las tareas especificadas. Con pocas líneas dentro del mismo código en Python, se puede incluir el servidor web con el interfaz de acceso al Alfabot. |

Tabla 2.3: Selección del servidor web para el desarrollo de la aplicación.

En base a esta información, LAMP queda totalmente descartado ya que debido a su complejidad, es la que más complicaciones podría traer. Node.JS podría ser una buena opción al trabajar con distintos lenguajes, pero teniendo Django y Flask que están desarrollados bajo lenguaje Python, son las dos opciones más lógicas a elegir.

Dado que la aplicación web únicamente pretende contener los controles del robot a los que accede un usuario remoto, la opción elegida ha sido Flask, ya que permite crear el sitio web desde su inicio indicando todo el contenido deseado, sin tener archivos de más que resulten innecesarios.

2.5.4. Conclusión

Todos los elementos software y hardware de este proyecto se resumen en la Tabla 2.4.

| Producto | S/H | Descripción |
|----------|----------|----------------------------------------------------------------------------------------------------|
| Raspbian | Software | Es el sistema operativo de la Raspberry Pi y del servidor web |
| Python | Software | Lenguaje de programación para el desarrollo de la aplicación web y el código de control del robot. |

| | | |
|-----------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------|
| Flask | Software | <i>Microframework</i> de Python implementado como servidor del sitio web. |
| CSS y HTML | Software | Lenguajes de programación para el desarrollo de la web del lado del cliente. |
| Router <i>gateway</i> | Hardware | La puerta de enlace, <i>gateway</i> , actúa de interfaz de conexión entre el robot y el dispositivo del usuario. |
| Raspberry | Hardware | Microcontrolador que funciona como unidad de control del Alphabot, y a la vez, como computadora donde se aloja la aplicación web. |
| Alphabot | Hardware | Robot móvil seleccionado para la implementación de su control remoto, junto con todos sus sensores y actuadores. |
| Adafruit DHT11 | Hardware | Sensor añadido al conjunto del Alphabot para la lectura de temperatura y humedad ambiente. |

Tabla 2.4: Software y hardware que componen el proyecto.

Capítulo 3

Diseño del control remoto y desarrollo

3.1. Introducción

En este capítulo se detalla el diseño y desarrollo del control remoto del robot móvil Alfabot, en base a las decisiones sobre los componentes y tecnologías tomadas en el capítulo anterior.

Éste comienza explicando el diseño del módulo de control remoto con todos sus elementos e interconexiones (Fig. 3.1), entrando más a fondo en el desarrollo de cada elemento y sus programas de hardware en las secciones siguientes.

3.2. Entorno del control remoto del robot

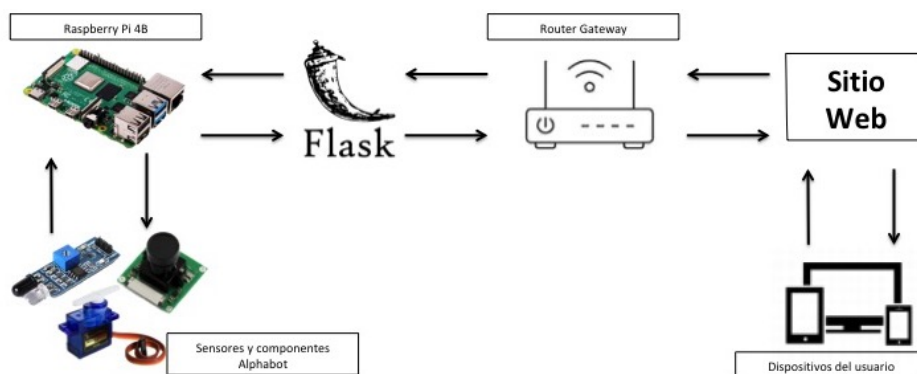


Figura 3.1: Arquitectura de la plataforma desarrollada.

La plataforma consta del Alphabot, que contiene una placa (PCB) electrónica que hace las funciones de interfaz entre los sensores y actuadores, y la unidad de proceso Raspberry Pi. El trasvase de información entre el usuario remoto y el robot se realiza inalámbricamente a través de una red local, usando un router. Tanto la página web como el servidor, están alojados en la Raspberry Pi, la unidad de proceso del Alphabot.

La Raspberry Pi controla todos los elementos físicos que permiten el movimiento del Alphabot: dos sensores infrarrojos, dos fotodetectores, los motores DC de las ruedas, los servomotores de la cámara y la adquisición de imágenes. La interfaz Wi-Fi le permite conectarse inalámbricamente y acceder a Internet. El control remoto y el acceso al servidor web, sin embargo, sólo puede realizarse por acceso local con direccionamiento IP.

El servidor se implementa mediante la librería Flask de Python, que con las funciones específicas, al arrancar inicia el servidor y mantiene el trasvase de datos con el usuario remoto cuando accede a la web de la aplicación a través del navegador (Fig. 3.2).



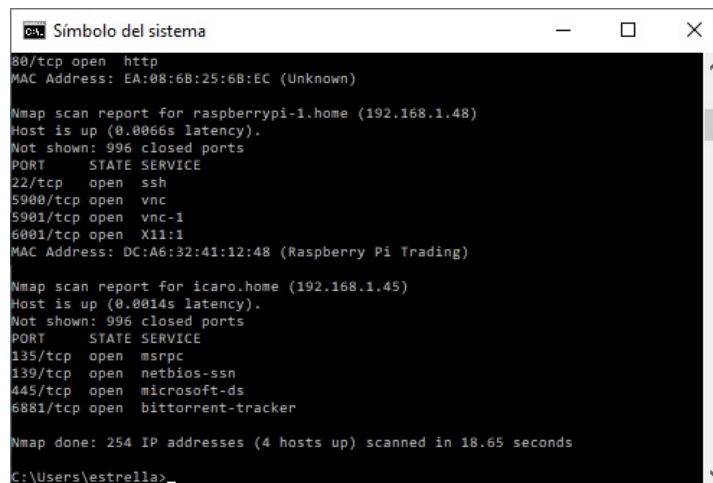
Figura 3.2: Página de inicio.

La aplicación contiene los controles para el movimiento manual remoto del Alphabot, controla la cámara Pi y presenta los datos de los sensores infrarrojos y de temperatura y humedad. El usuario puede acceder desde cualquier ordenador conectado en la misma red.

3.3. Configuración de red

Dado que Raspì, el sistema operativo, también permite direccionamiento IP, al iniciar el servidor Flask (comando @route), el modo de trabajo es 'localhost' con dirección IP privada 192.168.1.48 [75].

El acceso a Internet se hace inalámbricamente a través de un router Wi-Fi con IP pública, de manera que éste crea una red doméstica local asignando IPs privadas tanto a la Raspberry Pi como a los dispositivos usuario locales que quieran acceder a la misma (Fig. 3.3).



```

Símbolo del sistema
80/tcp open  http
MAC Address: EA:08:6B:25:6B:EC (Unknown)

Nmap scan report for raspberrypi-1.home (192.168.1.48)
Host is up (0.0066s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
5900/tcp   open  vnc
5901/tcp   open  vnc-1
6001/tcp   open  X11:1
MAC Address: DC:A6:32:41:12:48 (Raspberry Pi Trading)

Nmap scan report for icaro.home (192.168.1.45)
Host is up (0.0014s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
6881/tcp    open  bittorrent-tracker

Nmap done: 254 IP addresses (4 hosts up) scanned in 18.65 seconds
C:\Users\estrella>

```

Figura 3.3: IPs asignadas por el enrutador.

En este sentido, el direccionamiento del enrutador es dinámico, lo que significa que se asignan diferentes IPs a los dispositivos cada vez que se enciende y apaga el enrutador, provocando que el usuario de la red haya de cambiar la dirección IP en el navegador si quiere acceder a la página web.

3.4. Configuración básica de la unidad de proceso

El primer paso para poner en marcha la Raspberry Pi, es seguir el tutorial para instalar el asistente NOOBS (*New Out Of the Box Software*) que permitirá instalar el sistema operativo RasPi en una tarjeta uSD durante el primer proceso de arranque [76].

Cuando el sistema está instalado, se requieren unas configuraciones básicas para desarrollar la aplicación del Alfabot. Dado que el robot está diseñado para moverse con las ruedas y explorar, durante el proceso de desarrollo y depuración del código, es más cómodo eliminar todo tipo de cableado externo incluyendo: alimentador,

teclado, ratón y monitor. Por lo que el acceso remoto también se usa en las tareas de programación. Para que esto sea posible, se necesita por un lado poder lanzar órdenes en la línea de comandos de la Raspberry, y por otro, acceder a su escritorio para poder ejecutar aplicaciones y editar código remotamente.

Para lo primero, una posibilidad es el acceso SSH, que sigue un modelo cliente-servidor. Aquí, el cliente solicita una petición al servidor, el cual autentifica e inicia el entorno Shell. Es posible conectar varios clientes a un mismo servidor e incluye mecanismos de seguridad de encriptación de paquetes que no pueden ser descifrados en caso de ser interceptados.

Existen multitud de aplicaciones que se pueden instalar en PC remoto para conectarse con la Raspberry a modo de cliente SSH. Para este trabajo, se ha utilizado Putty (Fig. 3.4). Instalada la aplicación, la conexión se realiza introduciendo la dirección IP de la Raspberry Pi e indicando el puerto TCP 22. Al establecer la conexión, hay que introducir el usuario y contraseña que hayan configurados como usuario administrador de la Raspberry Pi.

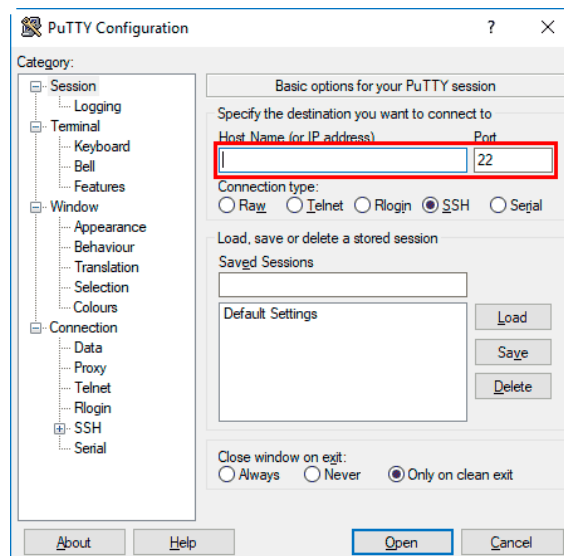


Figura 3.4: Putty.

Para el control del escritorio de manera remota, existe la aplicación VNC (*Virtual Network Computing*, Fig. 3.6), que a diferencia de SSH, permite abrir las aplicaciones de Raspbian al completo. La desventaja es que al necesitar transmitir una gran cantidad de datos al enviar señal de video, el comportamiento es más lento, todo y que facilita los cambios en los programas Python del control del robot o de la aplicación web. VNC se basa en el protocolo RFB (*Remote Frame Buffer*). Es capaz de transmitir señal de video, y los eventos de ratón y teclado entre varios ordenadores.

Ambas opciones, SSH y VNC, requieren activar la opción correspondiente accediendo al menú de configuración de Raspberry Pi a través de la barra de inicio (Fig. 3.5).

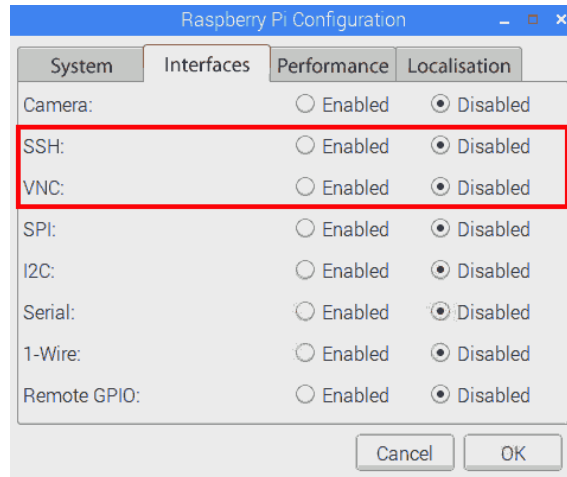


Figura 3.5: Opciones de interfaces de Raspberry Pi.

En el caso del servidor VNC, es necesario instalar VNCServer en el lado de la Raspberry, mientras que en el ordenador, tablet o móvil del cliente remoto, se debe instalar RealVNC. En este caso, el proceso de conexión remoto es bastante similar a SSH. Una vez instalados y ejecutados los programas, la conexión se realiza indicando la dirección IP de la Raspberry Pi, donde también habrá que introducir usuario y contraseña (Fig. 3.6). Realizada la conexión, la pantalla de RealVNC del cliente remoto mostrará el aspecto de la Figura 3.7.

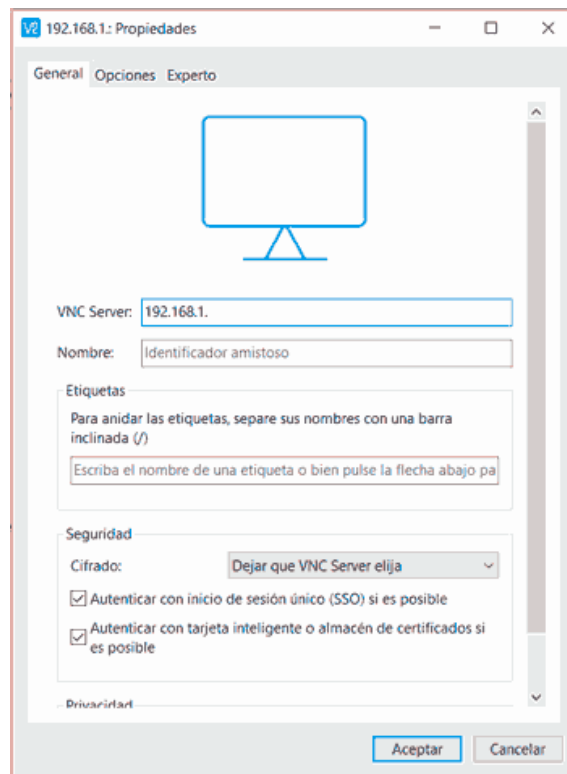


Figura 3.6: Configuración para conexión escritorio remoto.



Figura 3.7: Conexión al escritorio Raspbian.

Aunque este proyecto sólo utiliza el modo local, es posible conectarse desde Internet fuera de la red si se mapean adecuadamente los puertos del router.

3.5. Diseño de la interfaz web en localhost

La interfaz web es el entorno por el que el usuario envía los comandos, accede a la información del Alfabot, lee los datos que provienen de los sensores (temperatura, humedad y sensor de objetos) y controla tanto los movimientos del robot como la cámara para realizar fotos y vídeos.

Para implementarla, Flask combina diferentes tipos de archivos para gestionar el formato y el estilo de la página, además de la programación con el código. En este sentido, la aplicación se divide en dos niveles: uno en primer plano, denominado *Front-End*, con los objetos de la interfície de la página web, y otro en segundo plano, *Back-End*, con las funciones que se ejecutan durante la navegación y la interacción.

En este sentido, cuando el usuario desea realizar una acción - por ejemplo, una foto con la cámara o girar el robot a la izquierda - el plano HTML del *Front-End* envía la petición para que el *Back-End* la procese y éste envíe las señales a la placa controladora (Fig. 3.8). Pero, por otro lado, cuando los sensores actualizan su estado - por ejemplo, detección de un obstáculo o actualización del sensor de temperatura/humedad - la Raspberry enviará los datos a través del *Back-End* hacia el *Front-End*, para actualizar la página web HTML. De este modo, el *Back-End* se ejecuta siempre en segundo plano de manera permanente y transparente a la plataforma, mientras que el *Front-End* actualiza los valores cada vez que el usuario realiza una acción con los controles o refresca la página.

El código del *Back-End* se encuentra en el Capítulo 1 de los Anexos, mientras que los códigos del *Front-End* están repartidos entre el Capítulo 2 y 3.

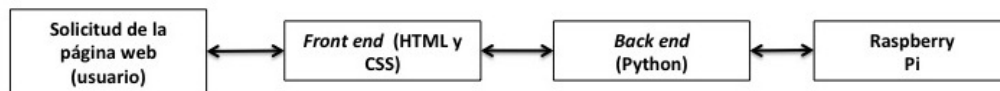


Figura 3.8: Flujo de la información entre módulos de la aplicación web

3.5.1. *Front-End*

El Front-End incluye los archivos en lenguaje HTML (*Hyper Text Mark-up Language*) con la estructura de las páginas web, el archivo de estilos CSS (*Cascade Style Sheets*) y el Javascript que implementa los scripts para la implementación de botones, controles y cuadros de diálogo, entre otros. En definitiva, gestiona los elementos de la web por el que el usuario interactúa con el robot.

Para este trabajo, se ha diseñado una interfaz web sencilla, minimalista y fácil de usar con un contenido receptivo para la interacción de los usuarios, para que se adapte al tamaño y resolución de pantalla de dispositivos que éste puede llegar a usar, como ordenadores portátiles, teléfonos móviles o tablets, entre otros. Esta experiencia de navegación se ha logrado activando la propiedad *'flex'* del fichero CCS, que establece un tamaño dinámico flexible de los elementos de la web. El fichero HTML llama al de CSS mediante un link a una hoja de estilos externa (*'stylesheet'*), referenciando el archivo que se quiere añadir, y la ruta a éste.

A continuación se explicarán los aspectos a considerar de los códigos del *front-end* que permiten que la interfaz se vea tal y como se ve.

En la página principal (Fig. 3.2) se ha fijado una imagen del Mars Pathfinder como fondo semitransparente. Para ello, se ha usado la propiedad de CSS: *'opacity'*. El menú con el que acceder a las diferentes secciones del sitio web se ha logrado con la etiqueta HTML ``, que significa *'unordered list'*, y `<nav>`, que identifica un elemento de navegación como es un menú. Se han referenciado las dos etiquetas, como clase *'navegación'* y *'menú'*, para luego poder referenciarlas desde el fichero CSS y darle el estilo que se desee al menú, y al aspecto de los elementos de la lista. Cada sección se ha creado con una etiqueta ``, de *'list'*, y referenciando en cada una, mediante el atributo *'href'*, la ubicación del fichero HTML creado para cada sección.

Los botones de las diferentes secciones de la aplicación web, como por ejemplo los del control del movimiento del robot (Fig. 3.9), se han creado con el elemento HTML `<a>`, que crea enlaces a otros archivos junto con el atributo *'href'*, mencionado anteriormente. Con él se llama a la función deseada dentro del código Python del servidor

web. Si por ejemplo el botón HTML llama al archivo `’/Movimiento/Forward’`, estará abriendo el comando `’@app.route/Movimiento/Forward’` donde se encuentra la acción deseada a realizar mediante ese botón. Para darle el estilo deseado, se han denominado todos con la clase `’button’` para referenciarlos desde el fichero CSS.

Por otro lado, para mostrar los datos de temperatura y humedad (Fig. 3.10), se ha utilizado JustGage, un elemento de JavaScript que genera y anima indicadores gráficos, y se ayuda de la librería Raphaël para crear el dibujo vectorial [77], refiriendo cada gráfico con un *id* que se le ha asignado en CSS, para diseñarlo como se desee: tamaño, color, entre otras propiedades.

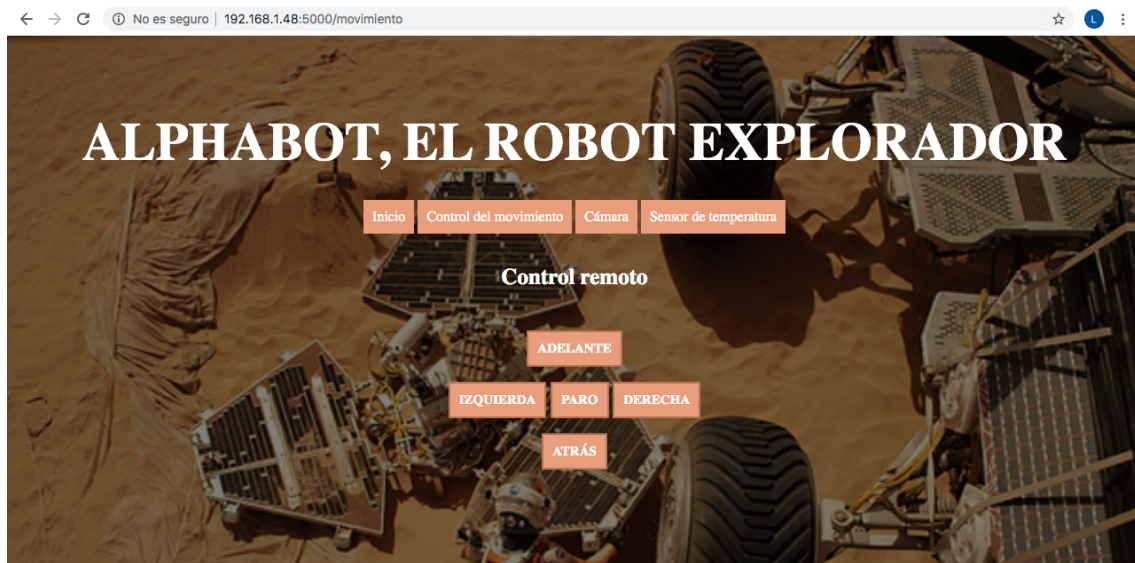


Figura 3.9: Sección para el control remoto de movimientos.



Figura 3.10: Sección que muestra datos de los sensores.

3.5.2. *Back-End*

En el segundo plano, *back-end*, se ejecuta el código encargado tanto de manejar y controlar los componentes y movimientos del robot, como el servidor Flask para el sitio web.

Este programa está alojado en la plataforma Raspberry Pi y realiza las acciones según el diagrama de bloques de la Figura 3.15. Aunque consta de un solo fichero, se encarga de leer los sensores, gestionar los actuadores para realizar el movimiento, manejar la cámara cuando se solicita, y con todo ello, gestionar la transferencia de datos de los componentes mencionados dependiendo del grado de interacción solicitada por el usuario.

Para que la Raspberry gestione los componentes del Alphabot, se han de conectar los GPIO mediante los jumpers marcados en la Figura 3.11. Al encender la plataforma del Alphabot, la Raspberry Pi abre su conexión Wi-Fi con el *gateway*, inicializa las variables de los motores, sensores y cámara, asignándole los puertos GPIO a los que están conectados, y por último pone en marcha el servidor web.

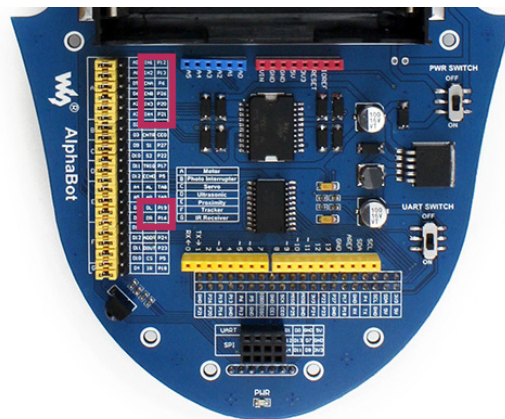


Figura 3.11: Mapa de los pines del Alphabot [44].

En cuanto al movimiento del Alphabot, la Tabla 3.1 muestra los terminales de la controladora que se usan para este fin junto con su correspondencia con los pines de salida del conector GPIO (Fig. 3.12).

Los pines ENA y ENB habilitan el puente H del driver de potencia L298P cuando hay que mover el robot. Para ello, los puertos P12-P13, que corresponden al motor izquierdo, y P20-P21 al derecho, han de activarse según se indica en la Tabla 3.2 para conseguir el patrón de movimiento deseado.

| Interfaz | Raspberry Pi |
|----------|--------------|
| IN1 | P12 |
| IN2 | P13 |
| ENA | P6 |
| IN3 | P20 |
| IN4 | P21 |
| ENB | P26 |

Tabla 3.1: Interfaces del Alphabot y su correspondencia respecto los puertos GPIO de la Raspberry Pi.

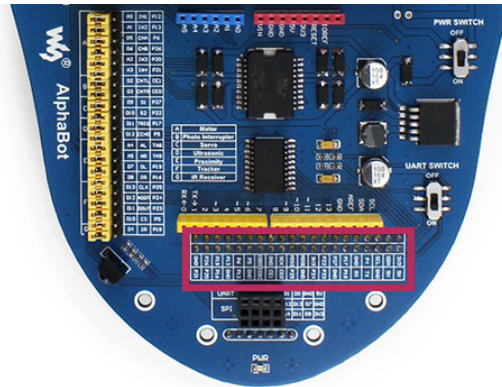


Figura 3.12: Mapa de los puertos GPIO de la RPi en el Alphabot.

| IN1 | IN2 | IN3 | IN4 | Descripción |
|-----|-----|-----|-----|-----------------------|
| 1 | 0 | 0 | 1 | Motores hacia delante |
| 0 | 1 | 1 | 0 | Motores hacia atrás |
| 0 | 0 | 0 | 1 | Giro a la izquierda |
| 1 | 0 | 0 | 0 | Giro a la derecha |
| 0 | 0 | 0 | 0 | Stop |

Tabla 3.2: Control teórico de los motores del AlphaBot.

El estado de los sensores infrarrojos para la detección de obstáculos, se lee a través de los terminales P19 (izquierdo) y P16 (derecho) del conector GPIO. Éstos están conectados físicamente a la controladora, en los puertos, DL y DR respectivamente, y se tienen que configurar como entradas digitales. Por otro lado, la cámara se conecta a su propia interfaz en el Alphabot, número 4 de la Figura 2.25, y por el otro lado del cable, al puerto del módulo de la cámara de la Raspberry Pi (Fig. 3.13).

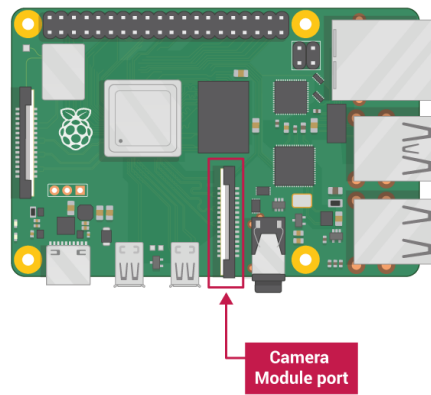


Figura 3.13: Puerto del módulo de la cámara [78].

El sensor DHT11, en cambio, consta de tres pines: uno de señal, otro de alimentación y el último de *ground*. El pin de señal se ha de conectar a un puerto GPIO. En este caso se ha elegido el puerto GPIO4, junto con la alimentación de 5V y tierra (Fig. 3.14).

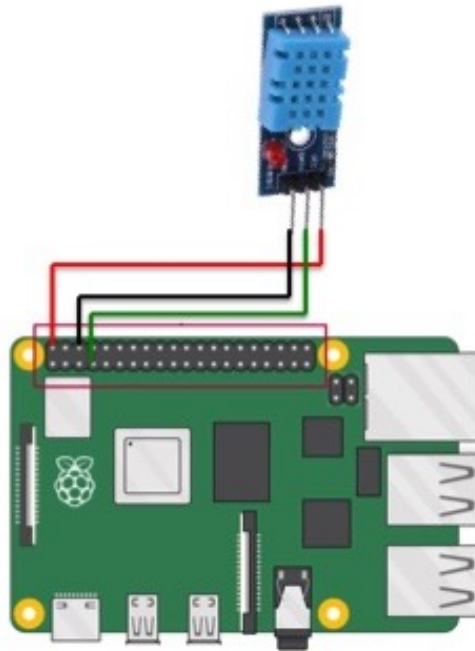


Figura 3.14: Conexión del sensor DHT11.

El funcionamiento del código implementado en el *Back-End* comienza importando las librerías necesarias para implementar el servidor Flask, usar los puertos GPIO de la Raspberry, la cámara, el sensor DHT11 y todos los periféricos del Alphabot utilizados (sensor de obstáculos, driver L298P, entre otros). Algunas de estas libre-

rías son *'Adafruit_DHT'*, para el sensor de temperatura y humedad, *'piCamera'*, para la cámara, y *'RPi.GPIO'*, para la conexión de los puertos, entre otras. Las bibliotecas relacionadas con el Alphabot en sí, junto con sus componentes, sin tener en cuenta el sensor de temperatura ajeno añadido, están disponibles en la web de Waveshare [79].

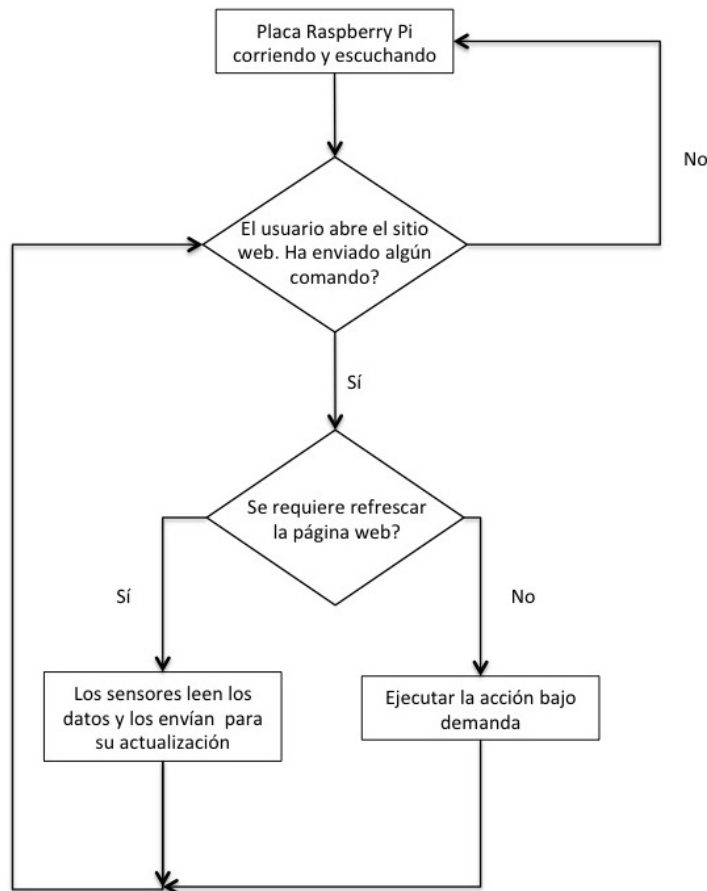


Figura 3.15: Diagrama de flujo simplificado del trabajo de la Raspberry Pi.

A continuación se inicializan todas las variables de estos periféricos y permanece a la espera de cualquier solicitud proveniente del *Front-End* realizada por el usuario. Las solicitudes del usuario que se han contemplado para este proyecto pueden ser de tres tipos:

- Solicitud de movimiento del robot
- Solicitud de capturar fotos y/o filmar vídeos
- Solicitud de conocer los datos leídos por los sensores de obstáculos, temperatura y humedad

Las dos primeras se consideran acciones de escritura donde el usuario quiere realizar una acción sobre el Alfabot, mientras que la tercera se considera acción de lectura para conocer el estado del Alfabot.

Al recibir una petición del *Front-End*, el *Back-End* identifica si ésta es de lectura o escritura, y siempre que esté conectado a red, el Alfabot ejecutará la acción correspondiente. Si es de escritura - girar a derecha o izquierda, moverse adelante o atrás, o realizar instantáneas con la cámara -, cada botón HTML asociado, referencia la acción deseada a realizar llamando al fichero .py que tiene definidas todas las acciones incorporadas en la aplicación web. Si por ejemplo se ejecuta el comando de 'Tomar foto' en la web, ese botón contiene el enlace '`href="/Camera/Take"`', que llama a '`@app.route("/Camera/Take")`', donde dentro se encuentra una acción definida de la siguiente manera: '`def action(Camera, Take)`', con todo lo necesario para realizar la tarea y ejecutarla.

En cambio, si el *Back-End* recibe una petición de lectura - sensor de temperatura/humedad y/u obstáculos -, realizará una llamada a una acción definida en el código, igual que si fuera de escritura, pero en este caso, realiza una lectura sobre el sensor de la plataforma especificado y refresca la pantalla con los datos nuevos leídos gracias a la librería '`render_template`' de Flask, la cual graba los valores que se deseen obtenidos por el código en una matriz llamada '`templateData`', y los devuelve al fichero HTML que se especifique para mostrarlos.

Al cumplir con la solicitud del usuario, vuelve al punto de espera para ejecutar algún comando.

3.6. Conclusión

Este capítulo ha cubierto el control hardware de los diferentes componentes de la plataforma robótica y el diseño de la web. Hasta el punto de redacción de esta memoria, el sistema descrito permite un funcionamiento básico donde la plataforma recibe peticiones vía web de un usuario remoto, para mover el robot, leer el estado de sus sensores, guardar fotos y grabar vídeos con la cámara.

La Raspberry obtiene su IP y un subdominio de red para que la comunicación entre usuario y unidad de proceso vía Internet sea posible (Secciones 3.3 y 3.4). La plataforma esta dividida en dos partes: un *Front-End* con el interfaz web que recoge las instancias, y que procesa los datos y peticiones de usuario, y el código python en el *Back-End* que realiza las acciones sobre el Alfabot.

Capítulo 4

Resultados: pruebas, puesta en marcha y funcionamiento

4.1. Introducción

Este capítulo tiene como objetivo validar la implementación del sistema del robot con el manejo remoto. Para ello, se muestra la capacidad de la aplicación web para conectarse con el robot, moverlo remotamente y obtener datos de sus sensores.

La primera sección (Sección 4.2) explica los pasos a realizar para ponerlo en funcionamiento para, a continuación, detallar el manejo de la plataforma a través de la web diseñada (Sección 4.3) con sus resultados.

4.2. Puesta en marcha

El primer y esencial paso para la puesta en marcha es encender el sistema y establecer la comunicación de la plataforma con la red. Para ello, después de ensamblar y verificar las partes del Alphabot hay que realizar los siguientes pasos:

- 1 Encendido. Tras activar el botón de encendido del Alphabot, el led PWR de la Raspberry se encenderá y el indicador ACT parpadeará indicando el proceso de arranque (Fig. 4.1). En su primer uso, es necesario conectar una pantalla al puerto HDMI o acceder mediante consola SSH o VNC para conectar la Raspberry a Internet por primera vez y conocer la IP asignada mediante Wi-Fi por el *router*. Una vez realizada la primera conexión, el sistema se conectará a la misma red automáticamente cada vez que se vuelva a encender.

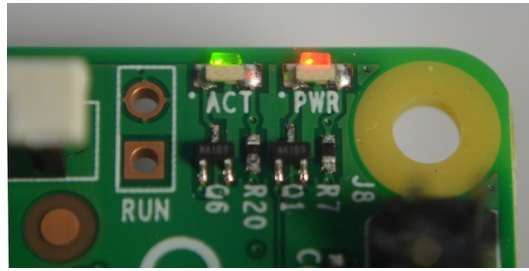


Figura 4.1: Leds ACT y PWR de Raspberry Pi.

- 2 Acceso SSH o VNC. Para ello, se debe seguir el proceso explicado en la Sección 3.4 poniendo la IP obtenida en el paso anterior.
- 3 Inicio de la aplicación servidor. En este punto hay que ejecutar la aplicación (webfinal.py) donde se aloja la web para que esté operativa y accesible por primera vez con la IP nueva. Una vez conocida la IP, el servidor se iniciará siempre que se encienda la Raspberry Pi, y con ella el Alfabot, sin necesidad de ejecutar el código.

Para que el dispositivo del usuario pueda acceder a la web debe estar conectado dentro de la red del enrutador.

Con los pasos anteriores, el sistema final obtenido será semejante al ilustrado en la Figura 4.2, variando las IP's de los elementos.

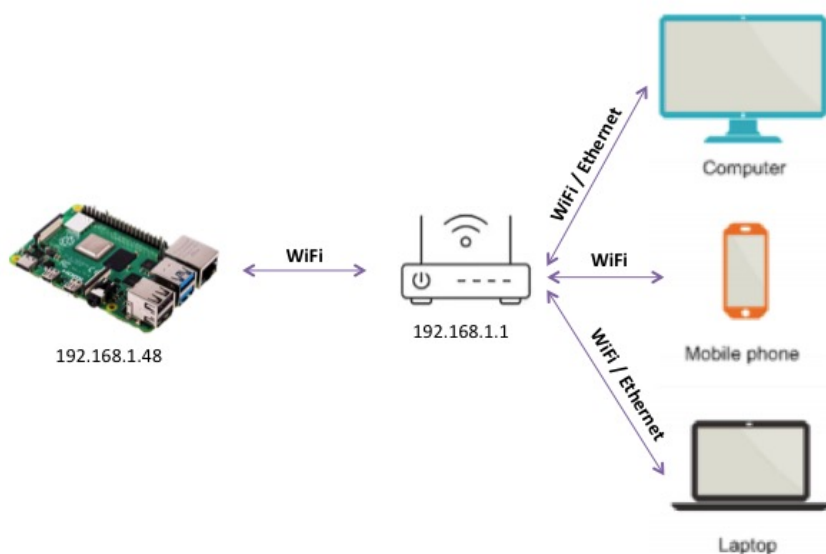


Figura 4.2: Esquema del sistema de red.

4.3. Resultados

Las secciones de la página web han acabado siendo repartidas en cuatro, aunque la sección de 'Inicio', que corresponde a la página principal, simplemente muestra la página de bienvenida. Las otras tres secciones, en cambio, son en las que el usuario puede realizar peticiones al Alhabot:

- *Control del movimiento.* Este menú despliega un mando de dirección por el que el usuario puede mover al robot (Fig. 3.9). Las opciones de movimiento son *adelante*, *atrás*, *derecha*, *izquierda* y *paro*. Para ello, el usuario debe clicar una sola vez encima del botón que quiera accionar, entonces el robot responderá rápidamente, y seguirá realizando esa acción todo el tiempo hasta que el cliente decida pararlo o cambiar de movimiento.
- *Cámara.* Este menú incluye las opciones de la cámara de vídeo (Fig. 4.3). Permite al usuario realizar fotos y vídeos mientras el robot está parado o en movimiento. Los archivos con las fotos y vídeos se guardan únicamente de forma directa en el escritorio Raspbian al que se puede acceder mediante el remoto VNCViewer (Fig. 4.4). El video en directo, hasta el momento, solo es posible si la Raspberry se encuentra conectada a un monitor. Los parámetros de la cámara, tales como resolución, frames por segundo, etc, son configurables a partir del código, donde salen especificados, pero no desde el lado del cliente una vez que haya accedido a la aplicación web.

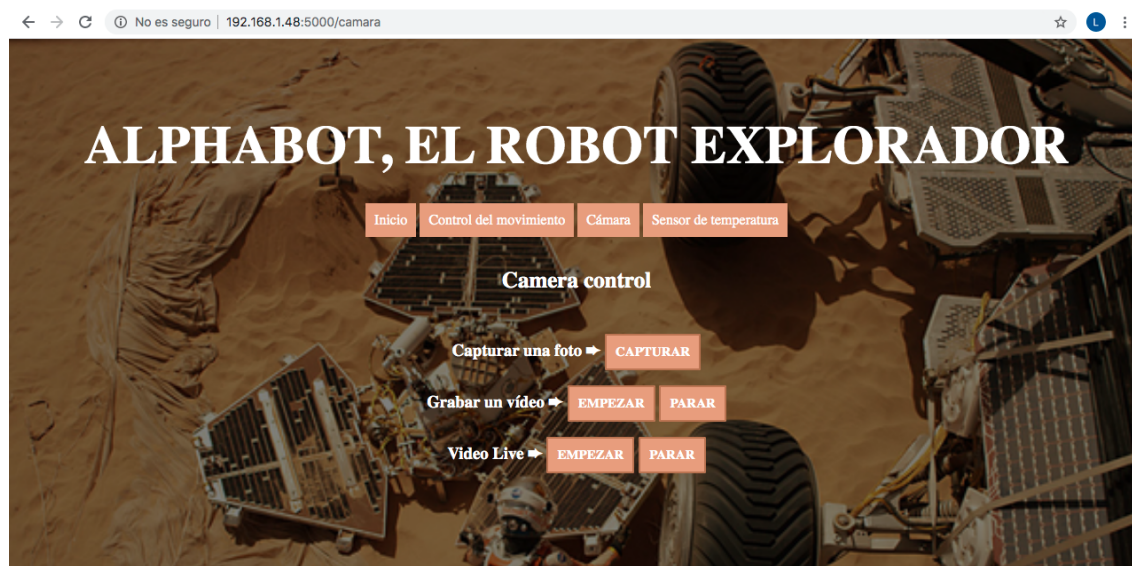


Figura 4.3: Sección de la cámara.

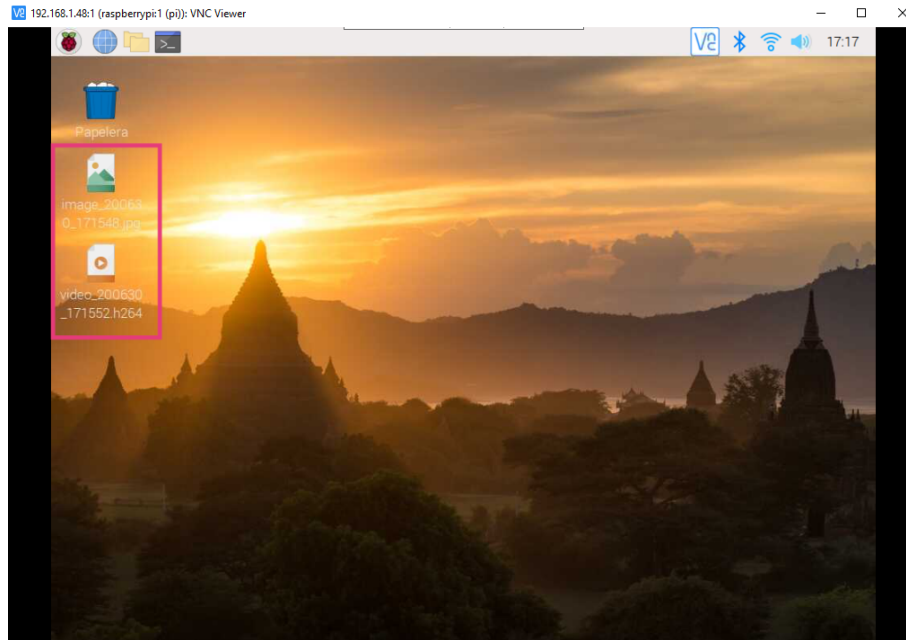


Figura 4.4: Escritorio de Raspbian con los archivos capturados y guardados automáticamente.

- *Sensores.* Esta sección muestra los valores de los sensores (Fig. 3.10). Por un lado, indica si hay un obstáculo delante del robot tanto por la parte izquierda como por la derecha, simplemente para el conocimiento del usuario, ya que el robot es capaz de esquivarlo mediante su propio código una vez detectado mediante la lectura constante de los pines DR y DL a lo largo de todo el código. Simplemente muestra '0' si no hay obstáculo y '1' en caso contrario, cada vez que el usuario refresca la página. Por el otro lado, cuando el sistema lee la temperatura y la humedad, se indica de manera gráfica. El código implementado no es capaz de acceder al historial de los valores obtenidos por los sensores, ni proceder a su descarga.

4.4. Conclusión

En este capítulo se ha verificado que el objetivo descrito en la Sección 1.2 ha sido cumplido correcta y completamente. Las diferentes partes de la página web han sido explicadas detallando el funcionamiento y alcance que tiene cada una.

Capítulo 5

Conclusiones y Trabajo futuro

5.1. Conclusiones

En este proyecto, y como primer prototipo, se ha desarrollado con éxito un ejemplo de robot explorador controlado remotamente. Para ello, era necesario determinar si el desarrollo de una web resultaba más conveniente que una interfície hombre-máquina. A lo largo del trabajo se ha demostrado que una interfície web es posible, incluso es aún mejor, ya que permite al usuario acceder a la gestión del robot desde cualquier plataforma, Windows, Mac o Linux. Por lo que es más estándar en relación al desarrollo de la interfície hombre-máquina, la cual requiere un aprendizaje considerable en el manejo de herramientas como Kivy, QT, entre otras, para luego desarrollar programas para las diferentes plataformas.

El hardware estudiado consta de una placa procesadora como unidad de control para el robot y una interfaz mediante un servidor web. Para llevar a cabo el control del robot, la plataforma lee la información sobre el estado del Alphabot a través de los sensores y componentes, y el código ejecuta los comandos para enviar las señales adecuadas a los actuadores.

Se ha creado un sitio web local en la misma placa de proceso, permitiendo al usuario una gestión del robot a nivel básico: manejo de la cámara, ejecución de movimientos y lectura de sensores: los infrarrojos para detección de obstáculos, y el DHT11 para medición de temperatura y humedad. El software correspondiente al sitio web, con el servidor, y el código de control del robot, han sido fusionados en un único programa, el cual tiene la escalabilidad suficiente para seguir añadiendo funciones que mejoren el código en un futuro sin que éste se vea afectado.

En el Capítulo 2 se expusieron los principios teóricos y que han servido de referencia para los robots móviles diseñados en la actualidad: tendencias de tecnologías y software para robots, junto con herramientas de desarrollo web con el fin de respaldar los robot móviles con aplicaciones muy útiles. También se entró en más detalle

en la robótica existente en el ámbito aeroespacial, por tratarse este proyecto de un pequeño prototipo para una posible misión de exploración. Seguidamente, se condujo a la selección de componentes de hardware, valorando sus pros y contras, que iban a formar la plataforma de control remoto de este estudio.

El Capítulo 3 mostró el diseño del sistema con la programación del hardware y software del robot, junto con el del sitio web, creando una arquitectura de red básica y fácil de transferir datos del robot al servidor.

En el Capítulo 4 se ha escrito un pequeño manual para ayudar al usuario final a configurar esta aplicación en caso de querer probar este proyecto, seguido de los resultados de éste.

5.2. Trabajo futuro

Actualmente, el campo de la robótica está teniendo un progreso muy destacable en muchos ámbitos, como la agricultura, el espacio, en coches autónomos, entre otros. En la agricultura, por ejemplo, se están usando tractores sin conductor que se ponen a trabajar la tierra sin intervención humana, mientras que en el ámbito espacial, el primer sector estratégico es la exploración planetaria.

En el caso de este proyecto, hay dos líneas generales que se pueden seguir a partir del trabajo realizado hasta el momento. Un grupo tendría que ver con la mejora de las funciones existentes en la interfície actual, incluyendo nuevas utilidades en la web, y el otro tendría que ver con el desarrollo de nuevas funcionalidades tanto software, como hardware, para el ámbito de aplicación del robot, que sería el aeroespacial.

5.2.1. Mejoras de la interfície

- Usar el **sensor de velocidad** para la lectura de este valor, y el **sensor de seguimiento de línea** para un posible guiado del Alphasbot por zonas marcadas. Son sensores que lleva incorporado el robot que por limitaciones de tiempo no se han llegado a utilizar.
- A partir de los encoders también sería posible **obtener otras variables físicas** a parte de la velocidad, como por ejemplo, el espacio recorrido y la aceleración, entre otras.
- Se podría implementar **una zona donde el usuario realice configuraciones de los sistemas que contiene el Alphasbot**. Por ejemplo, calibrar sensores y actuadores, o cambiar parámetros de la cámara como brillo, resolución, etc. Para ello, sería necesario sustituir algunos elementos de la plataforma por otros periféricos hardware capaces de realizar comunicaciones SPI, I2C, RS232, entre otras, permitiendo la trazabilidad de este tipo de datos entre los

módulos de sensores/actuadores y el procesador principal, ya que los implementados por el momento son cerrados que no permiten mucha flexibilidad en este sentido.

- Realizar funciones secundarias que permitan **gestionar los datos manejados por la plataforma** para lograr el tratamiento estadístico de las lecturas, listas e históricos obtenidos del sensor de temperatura y humedad DHT11, por ejemplo, o del sensor de velocidad para realizar una estadística de su velocidad y aceleración. Esto seguramente implicaría el uso de herramientas de gestión de bases de datos como MySQL replanteando el diseño de la aplicación software.
- Al mismo tiempo, incluir funciones y recursos para gestionar y **almanecar estos datos**. Como un menú para la gestión de archivos, o funciones para la transferencia de datos a un usuario remoto. Para ello sería necesario trabajar con otro tipo de protocolos de comunicación como MQTT (*Message Query Telemetry Transport*) o LWM2M (*LightWeight Machine-to-Machine*).
- Actualmente, el acceso remoto al Alfabot sólo se puede hacer a nivel local. Hacer que **un usuario acceda desde cualquier punto remoto por Internet**, requiere abandonar el mecanismo de IPs privadas de los router actuales para trabajar con DNS (*Direct Name Synthesis*). Una opción, sería contratar un servidor de red que ofreciese tal servicio o utilizar servicios como noIP.
- Adición de una pantalla a modo *display* que **informe al usuario del estado del Alfabot, de la acción que está ejecutando, y si se ha producido algún tipo de malfuncionamiento** en los módulos de éste.

5.2.2. Mejoras del software/hardware

- Un sensor interesante a **incluir** es el IMU (*Inertial Measurement Unit*). Puede medir la aceleración, velocidad de giro y campo magnético en los tres ejes. Esto ayudaría a obtener más información sobre la cinemática del robot, incluyendo otras variables como esfuerzo, fuerzas, momentos de inercia, entre otras.
- En cuanto a los actuadores, **sustituir las ruedas actuales por otras omnidireccionales y más robustas**, ofrecería más grados de libertad de movimiento del Alfabot. Con ellas, es posible hacer que el robot se mueva lateralmente si el eje trasero y delantero giran en direcciones opuestas, o movimientos más complejos si se combinan diferentes ejes. Del mismo modo, otra opción consiste en integrar brazos robóticos con elementos terminales, a modo de pinzas, para la adquisición de muestras del terreno explorado. Pero esto requeriría el uso de bases más flexibles a la hora de acoplar nuevos módulos, tanto mecánicos como hardware.
- La **incorporación de otros sensores y actuadores** como detectores de movimiento, sensores de luz, módulos de tiempo real para que el robot conozca

la fecha actual, sensores de inclinación para que determine su estabilidad y saber si ha volcado, entre otros.

- **Implementar otro protocolo de comunicación** entre usuario-robot que no dependa de una fuente de Internet Wi-Fi cercana. Estudiar la incorporación de un módulo más potente que sirva de intermediario en la comunicación mediante ondas de radio inalámbricas a una frecuencia UHF, parecido al sistema de comunicación que caracterizaba al Rover Sojourner.
- Unidades de computación como la Raspberry Pi, permiten instalar gran cantidad de **paquetes que permiten desarrollo de aplicaciones basadas en procesamiento avanzados de datos e Inteligencia Artificial (IA)**. Librerías Python como: *Numpy*, *SciKit*, *Pytorch*, *Keras*, *Pandas* o *TensorFlow*, entre otras, sirven para desarrollar aplicaciones de ML (*Machine Learning*), y se pueden utilizar para desarrollar aplicaciones avanzadas como vehículos autónomos, o procesamiento avanzado de datos, para una actividad como es la exploración espacial.
- En este trabajo, no se han estudiado aspectos relacionados con el consumo energético de la plataforma. No obstante, se intuye que la mayor parte de recursos energéticos los utilizan la Raspberry Pi, los actuadores y la comunicación inalámbrica Wi-Fi. Durante la realización del proyecto, se ha observado que la autonomía del robot es de 6 horas. Sin embargo, añadir aplicaciones de proceso de datos o Inteligencia Artificial, puede llevar a un incremento excesivo del consumo de la plataforma. Y por otro lado, se requiere de una fuente de alimentación que sea recargable sin necesidad de conexión eléctrica en el caso de misiones exploradoras. Por lo que **incorporar placas solares y sistemas de almacenamiento de energía extra** pueden ayudar a mantener la autonomía del robot. En este sentido, los sensores de luz, pueden ayudar a determinar cuando el sistema puede utilizar la energía solar o cargar las baterías, y automatizar la gestión de la misma.

Bibliografía

- [1] “Más Tecnología - NASA.” <https://rpp.pe/tecnologia/mas-tecnologia/nasa-explorar-el-universo>, [Accedido el 14/05/2020].
- [2] “Aeroespacial - La inversión en exploración espacial ascenderá a 260.000 millones de dólares en la próxima décadaNASA.” <https://actualidadaeroespacial.com/los-gobiernos-invertiran-260-000-millones-de-dolares-espacial>, [Accedido el 14/05/2020].
- [3] “Los robots no se ponen malos.” <https://www.larazon.es/economia/20200428/hjzkimc63bdg5az22yk2dqodmm.html/>, [Accedido el 14/05/2020].
- [4] “Robotica - Definición.” <https://robotica.wordpress.com/about/>, [Accedido el 24/04/2020].
- [5] “Planetario Alfa - ¿Para qué sirve la robótica?” <http://www.planetarioalfa.org.mx/?q=blog/%C2%BFpara-qu%C3%A9-sirve-la-rob%C3%B3tica>, [Accedido el 24/04/2020].
- [6] “Revista de Robot - ¿Qué es la robótica?” <https://revistaderobots.com/robots-y-robotica/que-es-la-robotica/>, [Accedido el 24/04/2020].
- [7] “NextComRobotics - Robótica.” <https://nextcomrobotics.wordpress.com/campo-de-aplicacion/aplicacion-de-la-robotica/>, [Accedido el 24/04/2020].
- [8] “Esneca - Qué es el brazo robótico y en qué industrias se emplea.” <https://www.esneca.com/blog/brazo-robotico-industrias/>, [Accedido el 24/04/2020].
- [9] “ABB - Robots industriales.” <https://new.abb.com/products/robotics/es/robots-industriales>, [Accedido el 24/04/2020].
- [10] “Coche global - Un ejército de 19.000 robots trabajan día y noche en las fábricas de coches.” <https://www.coheglobal.com/industria/un-ejercito-de-19-000-robots-trabajan-dia-y-noche-en-las-coches>, [Accedido el 14/05/2020].
- [11] “YoRobot - Robots para operaciones de riesgo.” <https://yorobot522.wordpress.com/category/robots-para-operaciones-de-riesgo/>, [Accedido el 24/04/2020].

- [12] “Xataka - PLUTO, un perro-robot autónomo diseñado para rescatar personas en túneles profundos, rocosos y sin mapear.” <https://www.xataka.com/robotica-e-ia/pluto-perro-robot-autonomo-disenado-para-rescatar-personas>, [Accedido el 24/04/2020].
- [13] Bambino, I., “Una introducción a los robots móviles,” tech. rep., 2008.
- [14] “Denoticias - Mercados globales de robots móviles autónomas de más de \$ 220 mil millones.” <https://www.denoticias.es/notas/mercados-globales-de-robots-moviles-autonomos-de-mas-de-220-mil>, [Accedido el 24/04/2020].
- [15] “iRobot - iRobot® Roomba® s9+.” https://tienda.irobot.es/irobot-roomba-s9/s955840.html?lang=es_ES&cgid=es&_ga=2.161600703.2035433686.1592347120-1407159038.1592347120, [Accedido el 09/06/2020].
- [16] “Boston Dynamics.” <https://www.bostondynamics.com/>, [Accedido el 18/06/2020].
- [17] “Xataka - Robomaster S1, análisis: el primer robot educativo de DJI es un arma para aprender a programar.” <https://www.xataka.com/analisis/dji-robomaster-s1analisis-caracteristicas-precio-especificaciones>, [Accedido el 09/06/2020].
- [18] “Xataka - DJI, la empresa líder en drones es china y no copia a nadie.” <https://www.xataka.com/drones/dji-la-empresa-lider-en-drones-es-china-y-no-copia-a-nadie>, [Accedido el 09/06/2020].
- [19] “ROBOT MILITAR - FEDOR, EL ROBOT SOLDADO RUSO.” <https://www.robotmilitar.org/fedor-el-robot-soldado-ruso/>, [Accedido el 09/06/2020].
- [20] “Montevideo - Rusia envía al espacio a Fedor, su primer robot humanoide.” <https://www.montevideo.com.uy/Ciencia-y-Tecnologia/Rusia-envia-al-espacio-a-Fedor-su-primer-robot-humanoide-uc727716>, [Accedido el 09/06/2020].
- [21] “National Geographic - Yuri Gagarin, el primer hombre en el espacio.” https://www.nationalgeographic.com.es/fotografia/foto-del-dia/yuri-gagarin-primer-hombre-espacio_10278, [Accedido el 18/06/2020].
- [22] “Sputnik 1.” https://es.wikipedia.org/wiki/Sputnik_1, [Accedido el 28/05/2020].
- [23] “Estación Espacial Internacional.” https://es.wikipedia.org/wiki/Estación_Espacial_Internacional, [Accedido el 30/05/2020].
- [24] “AstroAfición - La Estación Espacial Internacional o ISS.” <https://astroaficion.com/2016/12/19/estacion-espacial-internacional/>, [Accedido el 30/05/2020].

- [25] “Hacia el Espacio - Robots Espaciales.” <http://haciaelespacio.aem.gob.mx/revistadigital/articul.php?interior=39>, [Accedido el 31/05/2020].
- [26] “NASA - Una descripción de Rover Sojourner.” <https://mars.nasa.gov/MPF/rover/descrip.html>, [Accedido el 30/05/2020].
- [27] “Descripción del instrumento de la cámara móvil.” https://pdsimage2.wr.usgs.gov/data/mpfr-m-apxs-2-edr-v1.0/mprv_0001/document/rcinst.htm, [Accedido el 30/05/2020].
- [28] “NASA - Mars Microrover Power Subsystem.” <https://mars.nasa.gov/MPF/roverpwr/power.html>, [Accedido el 30/05/2020].
- [29] “Radiomódem.” <https://es.wikipedia.org/wiki/Radiomódem>, [Accedido el 30/05/2020].
- [30] “UHF.” <https://es.wikipedia.org/wiki/UHF>, [Accedido el 30/05/2020].
- [31] “NASA - Microrover Radios and Antennas.” <https://mars.nasa.gov/MPF/rovercom/itworks.html#rover-radio>, [Accedido el 30/05/2020].
- [32] “Alfa partícula espectrómetro de rayos X - Alpha particle X-ray spectrometer.” https://es.qwe.wiki/wiki/Alpha_particle_X-ray_spectrometer, [Accedido el 30/05/2020].
- [33] “Balón Cuadrado - Robot explorador Curiosity toma video en 360 grados de Marte.” <http://baloncuadrado.com/2019/02/11/robot-explorador-curiosity-toma-video-en-360-grados-de-marte/>, [Accedido el 08/03/2020].
- [34] “Behance - Multi Mission Radioisotope Thermal Generator / MMRTG.” <https://www.behance.net/gallery/14207437/Multi-Mission-Radioisotope-Thermal-Generator-MMRTG>, [Accedido el 18/06/2020].
- [35] “Real-time operating system.” https://en.wikipedia.org/wiki/Real-time_operating_system, [Accedido el 31/05/2020].
- [36] “NASA: Your smartphone is as smart as the Curiosity rover.” <https://www.computerworld.com/article/2505612/nasa--your-smartphone-is-as-smart-as-the-curiosity-rover.html>, [Accedido el 31/05/2020].
- [37] “Curiosity (rover).” [https://en.wikipedia.org/wiki/Curiosity_\(rover\)](https://en.wikipedia.org/wiki/Curiosity_(rover)), [Accedido el 31/05/2020].
- [38] “NASA - Mars Science Laboratory Telecommunications System Design.” https://descanso.jpl.nasa.gov/DPSummary/Descanso14_MSL_Telecom.pdf, [Accedido el 31/05/2020].

- [39] “Electra (radio).” [https://en.wikipedia.org/wiki/Electra_\(radio\)](https://en.wikipedia.org/wiki/Electra_(radio)), [Accedido el 31/05/2020].
- [40] “NASA - The Mars Rovers: Curiosity.” <https://spaceplace.nasa.gov/mars-curiosity/sp/>, [Accedido el 18/06/2020].
- [41] “Arturo Goga - Conoce las Camaras del Curiosity Rover.” <https://www.arturogoga.com/conoce-las-cmaras-de-curiosity-rover/>, [Accedido el 20/06/2020].
- [42] “Business Insider - Robots controlados de forma remota podrían sustituir a los astronautas en el futuro.” <https://www.businessinsider.es/robots-controlados-remotamente-sustituiran-astronautas-643179>, [Accedido el 31/05/2020].
- [43] “Canadarm.” <https://es.wikipedia.org/wiki/Canadarm>, [Accedido el 31/05/2020].
- [44] “AlphaBot User Manual.” <https://www.waveshare.com/w/upload/c/c7/AlphaBot-User-Manual.pdf>, [Accedido el 06/06/2020].
- [45] “Irene robótica - Componentes de un robot.” <https://sites.google.com/site/irenerobotica/4-robots-industriales>, [Accedido el 25/04/2020].
- [46] “Microcontrolador – qué es y para que sirve.” <https://hetpro-store.com/TUTORIALES/microcontrolador/>, [Accedido el 26/05/2020].
- [47] “Aprendiendo Arduino – Hardware Arduino para la Educación.” <https://aprendiendoarduino.wordpress.com/tag/microcontroladores/>, [Accedido el 27/05/2020].
- [48] “Xataka – Raspberry Pi 4 es oficial: una completa actualización con procesador Cortex-A72, hasta 4 GB de RAM y desde 35 dólares.” <https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica>, [Accedido el 27/05/2020].
- [49] “¿Qué es eso? ¿Es importante?” <https://catedu.gitbooks.io/raspberry-muy-basico/content/2-gpio.html>, [Accedido el 04/06/2020].
- [50] “Arduino UNO Rev3.” <https://store.arduino.cc/arduino-uno-rev3>, [Accedido el 18/06/2020].
- [51] “Arduino.” <https://crlpiban.wordpress.com/2017/09/06/arduino/>, [Accedido el 04/06/2020].
- [52] “CEAC - La importancia de los sensores en robótica.” <https://www.ceac.es/blog/la-importancia-de-los-sensores-en-robotica>, [Accedido el 25/04/2020].

- [53] “Robots Argentina - Sensores, conceptos generales.” http://robots-argentina.com.ar/Sensores_general.htm, [Accedido el 25/04/2020].
- [54] “Mecatrónica LATAM - Sensor de luz.” <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-luz/>, [Accedido el 25/04/2020].
- [55] “Factor Evolución - Sensor de luz.” <https://www.factor.mx/portal/base-de-conocimiento/sensor-de-luz/>, [Accedido el 26/04/2020].
- [56] “Keyence - ¿Qué es un sensor fotoeléctrico?” <https://www.keyence.com.mx/ss/products/sensor/sensorbasics/photoelectric/info/>, [Accedido el 26/04/2020].
- [57] “¿Qué tipos de sensores fotoeléctricos existen?” <https://www.contaval.es/que-tipos-de-sensores-fotoelectricos-existen/>, [Accedido el 31/05/2020].
- [58] “Play embedded - Detección de obstáculos con sensor IR y Arduino.” <https://www.playembedded.org/blog/detecting-obstacle-with-ir-sensor-and-arduino/>, [Accedido el 31/05/2020].
- [59] “Zona Maker - Sensor de ultrasonidos HC-SR04.” <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/ultrasonido-hc-sr04>, [Accedido el 26/04/2020].
- [60] “Hardwarelibre - HC-SR04: todo sobre el sensor de ultrasonidos.” <https://www.hwlibre.com/hc-sr04/>, [Accedido el 02/06/2020].
- [61] “Mundo Motor - Sensor De Temperatura: Características Y Funcionamiento.” https://www.mundodelmotor.net/sensor-de-temperatura/#sensor_de_temperatura, [Accedido el 26/04/2020].
- [62] “EcuRed - Sensor de Humedad.” https://www.ecured.cu/Sensor_de_Humedad, [Accedido el 26/04/2020].
- [63] “Programar fácil - Cómo utilizar el sensor DHT11 para medir la temperatura y humedad con Arduino.” <https://programarfácil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/#:~:text=Una%20de%20las%20ventajas%20que,m%C3%A1s%20protegidos%20frente%20al%20ruido.>, [Accedido el 18/06/2020].
- [64] “CLR - Tipos de encoders y aplicaciones en motores.” <https://clr.es/blog/es/tipos-de-encoders-aplicaciones-motores/>, [Accedido el 25/05/2020].
- [65] “Motor paso a paso ¿que es y como funciona?” <https://www.ingmecafenix.com/electricidad-industrial/motor-paso-a-paso/>, [Accedido el 26/05/2020].

- [66] “Prometec - Más sobre motores de corriente continua.” <https://www.prometec.net/hbridge/>, [Accedido el 20/06/2020].
- [67] “Todo lo que necesitas saber, antes de comprar un servomotor.” <https://www.contaval.es/lo-necesitas-saber-comprar-servomotor/>, [Accedido el 26/05/2020].
- [68] “SERVOS.” <https://catedu.gitbooks.io/alphabot/content/6-servos.html>, [Accedido el 09/06/2020].
- [69] “Kisiblog - 6 Communication Protocols Used by IoT.” <https://www.getkisi.com/blog/internet-of-things-communication-protocols>, [Accedido el 31/05/2020].
- [70] “LCM: Lightweight Communications and Marshalling.” <https://lcm-proj.github.io/>, [Accedido el 20/06/2020].
- [71] “Netcraft - Encuesta del servidor web de mayo de 2020.” <https://news.netcraft.com/archives/category/web-server-survey/>, [Accedido el 31/05/2020].
- [72] “LAMP (software bundle).” [https://en.wikipedia.org/wiki/LAMP_\(software_bundle\)](https://en.wikipedia.org/wiki/LAMP_(software_bundle)), [Accedido el 31/05/2020].
- [73] “MDN web docs - Introducción a Django.” <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introducción>, [Accedido el 31/05/2020].
- [74] “Python.” <https://es.wikipedia.org/wiki/Python>, [Accedido el 31/05/2020].
- [75] “Hostinger - ¿Qué es localhost?” <https://www.hostinger.es/tutoriales/que-es-localhost/>, [Accedido el 02/06/2020].
- [76] “Comenzando con Raspberry Pi.” <https://projects.raspberrypi.org/es-ES/projects/raspberry-pi-getting-started>, [Accedido el 09/06/2020].
- [77] “JustGage - ¿Qué es JustGage?” <https://toorshia.github.io/justgage/>, [Accedido el 18/06/2020].
- [78] “Comenzando con el módulo de la cámara.” <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/1>, [Accedido el 09/06/2020].
- [79] “Waveshare - Libraries Installation for RPi.” https://www.waveshare.com/wiki/Libraries_Installation_for_RPi, [Accedido el 18/06/2020].